ADAPTING SEMICONDUCTOR MEMORY TO SMALL MACHINE APPLICATIONS

By

ROBERT P. BILLEG, JR.

Bachelor of Science

University of Oklahoma

Norman, Oklahoma

1963

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
May, 1974

ADAPTING SEMICONDUCTOR MEMORY TO

SMALL MACHINE APPLICATIONS

Thesis Approved:

Edward L. Shreve
Thesis Adviser

Paul A. McCollum

Richard L. Cummins

D D Dunham
Dean of the Graduate College

PREFACE

This study was undertaken to present the techniques and design principles required for a complete semiconductor memory system. There are few, if any, documents that describe a semiconductor memory system and the subsystems that it contains. During my tenure in industry, I have been primarily concerned with the development of various semiconductor memory systems used in small peripheral and processor applications. Each new assignment requires some different techniques, but most applications use repetition of the same basic equations, same basic methods, and same basic needs. This thesis represents the documentation of this experience adapted for the most recent Read/Write and Read Only memories to be introduced within the industry. The techniques and methods will be applicable to many needs.

The author wishes to express appreciation to his major adviser, Doctor Edward Shreve, for his assistance and advice throughout the preparation of this paper. Appreciation is also extended to the members of the committee, for their assistance in the final preparation of the manuscript.

I also want to thank the members of the engineering staff at Honeywell Information Systems, Inc., Oklahoma City, for their assistance and in particular, Robert Bibles for his help in the preparation of the illustrations and tables. A note of thanks is also expressed to the Honeywell secretaries who typed the draft copies and to Dixie Jennings for the final typed manuscript.

A special gratitude to my wife, Sharon, and my four children for the many sacrifices when education demanded time that should have been theirs.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

CMOS	Complementary Metal Oxide Semiconductor
IC	Integrated Circuit
LSI	Large Scale Integration
MHS	Frequency abbreviation equal to 1×10^6 cycles per second, i.e., megahertz
MOS	Metal Oxide Semiconductor
MSI	Medium Scale Integration
RAM	Random Access Memory, often used in semiconductor terms for the Read/Write memory
ROM	Read Only Memory
SOS	Metal Oxide Semiconductor fabrication technique, Silicon on Sapphire
SSI	Small Scale Integration, commonly referred to as the IC gates, flip flops, and inverters
TTL	Transistor Transistor Logic
V_c	Capacitor Potential
V_{CC}	Bipolar technology power supply voltage
V_{CO}	Potential on a capacitor at the time of initialization
V_{DD}	Metal Oxide Semiconductor transistor source potential
V_{GG}	Metal Oxide Semiconductor transistor gate bias potential
V_{SS}	Metal Oxide Semiconductor transistor drain potential
ms	Milliseconds (10×10^{-3} seconds)

n-channel	Metal Oxide Semiconductor majority carrier designator, electrons
ns	Nanoseconds (10×10^{-9} seconds)
p-channel	Metal Oxide Semiconductor majority carrier designator, holes
pf	Picofarads (10×10^{-12} farads)
µfd	Microfarads (10×10^{-6} farads)
µsec	Microseconds (10×10^{-6} seconds)
Ω	Ohms

CHAPTER I

INTRODUCTION TO A SEMICONDUCTOR MEMORY

Format of a Memory Design

This paper represents the results of five years in designing industrial and commercial processor memory systems. The accumulation, the techniques, and the criteria of this experience is offered in this thesis. Chapter I presents the historical background and the architecture of a semiconductor processor memory. The remaining chapters present the design techniques that can be used for the design and manufacturing of a memory. The memory is presented as a subsystem of a typical small processor. Shown in Figure 1 in a flowchart format, are the design steps that this paper outlines.

History

For many years the magnetic core memory has dominated both large and medium computer memory applications. Only within the past two years has it been seriously challenged. This challenge is the more economical, faster access and cycle time and higher package density, monolithic semiconductor memory.

Most non-magnetic memory used the Metal Oxide Semiconductor (MOS) technology. The first applications of the MOS memories were not processor oriented. The majority were used in computer peripheral equipment where the memory was organized serially. This was because the shift

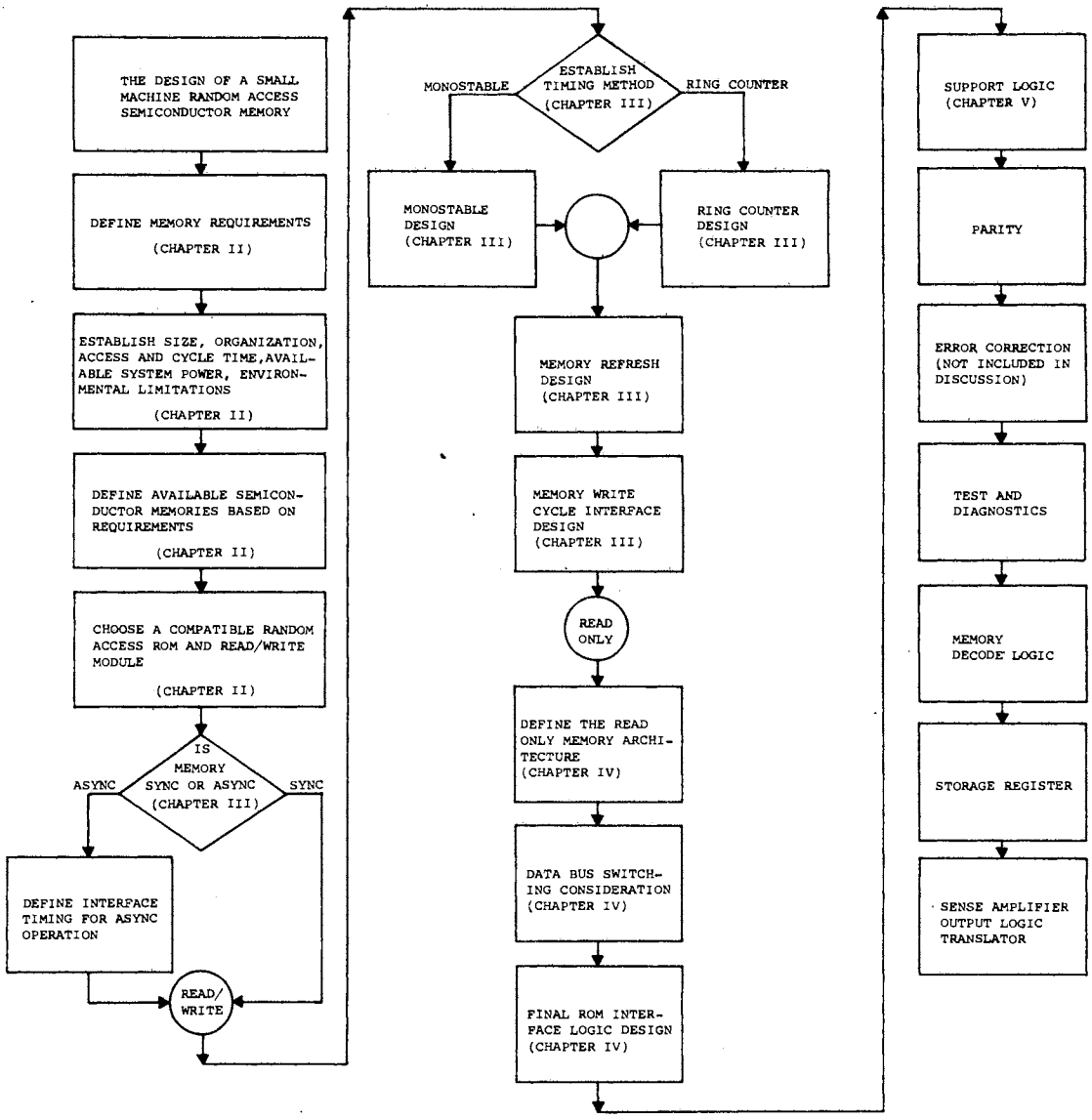


Figure 1. A Flow Chart Representative of the Design Process

register, making use of the capacitance nodal storage property of the MOS device, was the first device that could be economically produced. Memory sizes were restricted to a few thousand bits, with a maximum clock frequency of 1 MHZ. Individual package size was confined to a maximum of 100 bits. This capacity was later expanded to one and two thousand bits with a clock frequency exceeding 5 MHZ. However, the usefulness of the serial memory was limited due to the introduction of the semiconductor monolithic random access memory.

In 1967, the first random access semiconductor memory was made available as a commercial product. It was in the form of the Read Only and was the first true processor oriented semiconducted memory. In 1969, the semiconductor memory systems became practical with the introduction of the semiconductor random access Read/Write memory. The performance needs of processor applications could now be cost effective using a semiconductor memory. Once again the MOS technique was used first, because of the increased package density and its ability to store data on a MOS capacitance storage node. Individual monolithic IC sizes up to 1024 bits were available in early 1970. In the same period the bipolar Read/Write memory became available for system use where fast cycle and access times were required. Practical bipolar Read/Write memories do not now exceed 256 bits, even though a few applications, particularly military, do use the more expensive bipolar ROM's.

The use of semiconductor memory in small machine applications is becoming widely accepted because of the cost effectiveness of semiconductor memory opposed to the magnetic core memory. A few exceptions to this are the small applications found throughout the computer industry where a very large core memory (exceeding a few million words) has been

replaced by a semiconductor memory. However, the trend is definitely away from magnetic core and to the semiconductor memory. This is especially true as semiconductor memory becomes less expensive than core.

Present Day Usage

The memories that are being designed and used today are generally classified into three areas: (1) Serial Memories; (2) Read/Write, random access memory; and (3) Read Only, random access memory.

The serial memory consists of a series of shift registers and is not generally found in processor oriented machines. The serial memory is more often found in computer peripheral equipment such as line printers or Input/Output machines.

On the other hand, the Read/Write and Read/Only random access memory are found much more in the processor oriented machine as memory. Storage and retrieval are done on command. Opposed to this, the serial memory requires multiple bit time delays to access the desired memory location.

A Small Machines Architecture

A computer system, called a processor, is comprised of four interacting subsystems: (1) an arithmetic logic unit; (2) a control unit; (3) an input/output unit; and (4) a memory unit. Each unit is a separate entity in itself, but dependent on the other units to perform its intended function.

The purpose of this paper is to define, by example, the design criteria to make a semiconductor memory an interacting subsystem within a small computer system. Each chapter is divided into a portion of the memory subsystem.

As stated in the previous section, there are two basic memory architectures for processing machines, serial or random access. This paper will pursue the random access memory as this is the largest percentage of the usage.

The memory function can be asynchronous or synchronous with respect to processor operation. In synchronous operation the flow of address and storage information to and from memory is controlled by the arithmetic and control units master clock. In asynchronous operation the memory must communicate with the processor to determine a mutually agreeable time for data transfer to take place. Because most memories operate synchronously, thereby eliminating the need for external transfer control, this paper will concentrate on synchronous operation. Figure 2 is a general block diagram of a synchronous random access memory.

Often the performance and operation of a small processor requires both an alterable Read/Write and a non-alterable Read Only random access memory. This paper offers the engineer a worst case design outline for designing a small processors semiconductor memory. Many design techniques are applicable to large processor designs, but these within this text are directed primarily at memories whose capacities are less than 70,000 words.

The most common and economical memory configuration used in a small processor is to bus both the Read Only and the Read/Write memories on the same data and address bus. This calls for common data and address registers in the control and arithmetic units thereby reducing costs by eliminating duplication. Additional economy, and often physical packaging size, can be realized if the input and output data bus is mutliplexed. This in general, results in a need to reduce the memory

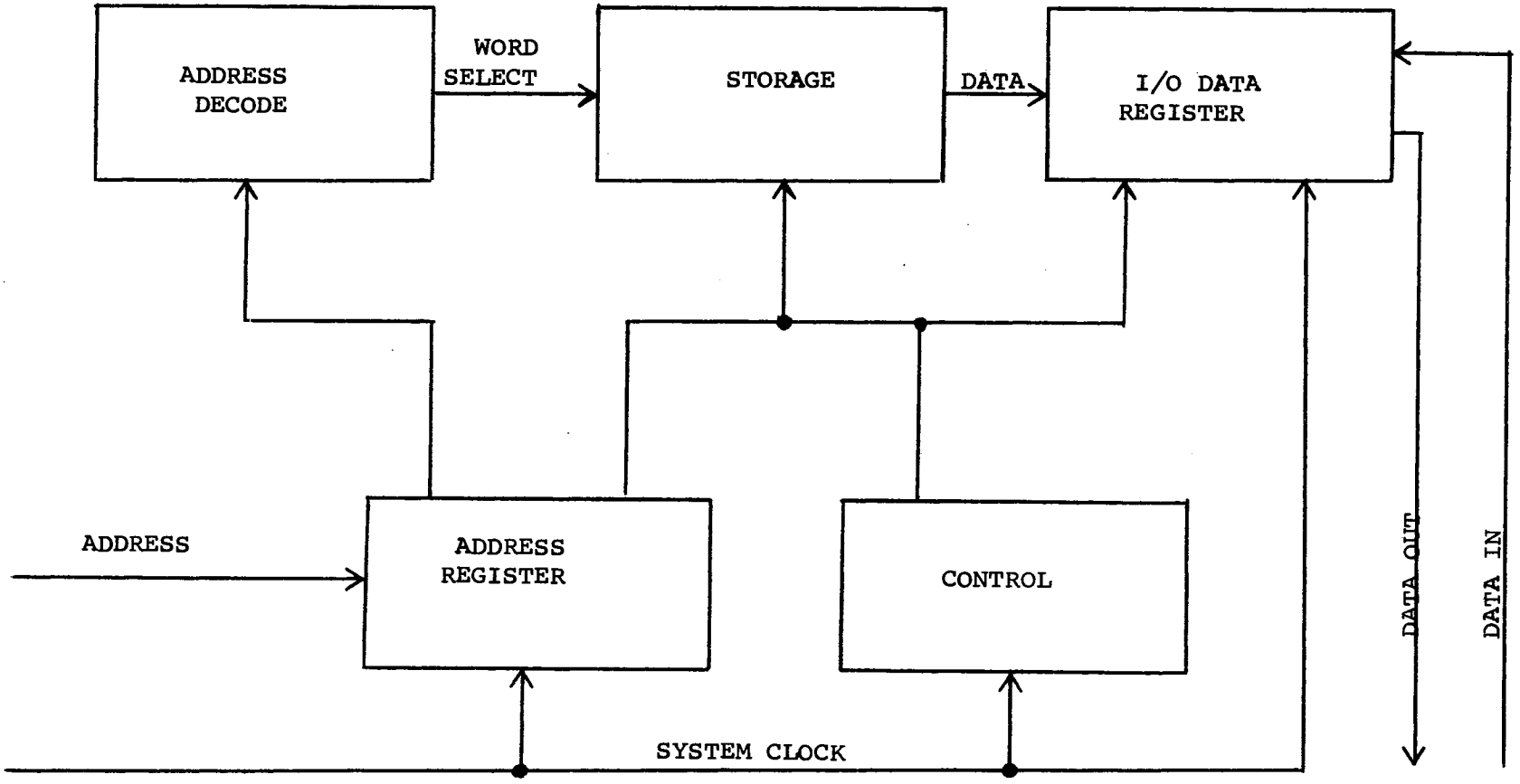


Figure 2. A Block Diagram of a Random Access Memory

storage elements cycle and access time, as time must be allotted for the Input/Output (I/O) buffer to switch states.

The memory unit itself is composed of nine subsystems each performing a separate task. A semiconductor memory is composed of: (1) address decode logic; (2) signal buffers; (3) control logic; (4) output buffers; (5) output register; (6) storage element; (7) parity; (8) refresh if required by the storage element; and (9) the timing interface logic. In all recently developed semiconductor memory integrated circuits, the address decode logic is built into the memory chip, thereby leaving the other 8 subsystems to be designed.

Chapter II details the design considerations and constraints when an integrated circuit memory is chosen. These criteria are outlined in Chapter II, but for now are listed to lend emphasis to the further discussions of this chapter. The design criteria are: (1) cost; (2) packaging; (3) performance; (4) storage capacity; (5) organization; (6) environmental; and (7) power supply available and required.

The following chapters are developed based on a Read/Write integrated circuit chosen for its recent announcement into the commercial field, and because of its storage capacity. It was also chosen for the degree of difficulty in implementing it into a processor's memory unit. The storage element is an integrated circuit, Read/Write semiconductor memory manufactured by Microsystems International Limited, of Ottawa, Canada. The memory element is typical of the 4096 bit Read/Write semiconductor memories being introduced to industry during the latter part of 1973 and the initial part of 1974. The integrated circuit is designated the MF 7112, and is packaged in a 22 pin dual in-line (DIP). The manufacturer's data sheet is included in the appendix.

The MF 7112 was also chosen as the example memory storage element because of its complexity in interfacing with the processor. The MF 7112 is complex because of the design techniques used to implement the internal storage cell.

A semiconductor memory can always be categorized as either static or dynamic. Static memory cells are cross-coupled bistable circuits wherein information is stored by one of the two stable states. Most static MOS memory cells use six transistors for each cell. Dynamic circuits use the absence or presence of charge on a capacitor to store information, typically with three or four transistors per cell. Almost all recently introduced 4096 bit Read/Write memory elements use only one transistor per storage cell. Since the capacitor that stores the charge has a leakage current, the stored information degrades slowly and therefore must be periodically refreshed. The MF 7112 is a dynamic memory requiring periodic refreshing.

A dynamic cell based on a two phase ratioless MOS device is shown in Figure 3. The cell is advantageous in the small area of silicon required per cell and the low power dissipation. The dynamic cell exhibits a low noise immunity and can be considered a serious disadvantage.

A three transistor cell is shown in Figure 4. This is the type cell used in the popular 1103, 1024 X 1 Read/Write memory. A feedback network is also shown in Figure 4. This network is used to refresh the storage node A. A dynamic cell requires continuous refresh to assure long term storage. Chapter III deals with memory refresh.

The three transistor and the one transistor cell (not shown) have been used exclusively in all new 4096 bit Read/Write memory designs. The one transistor cell holds an advantage because of the small amount of

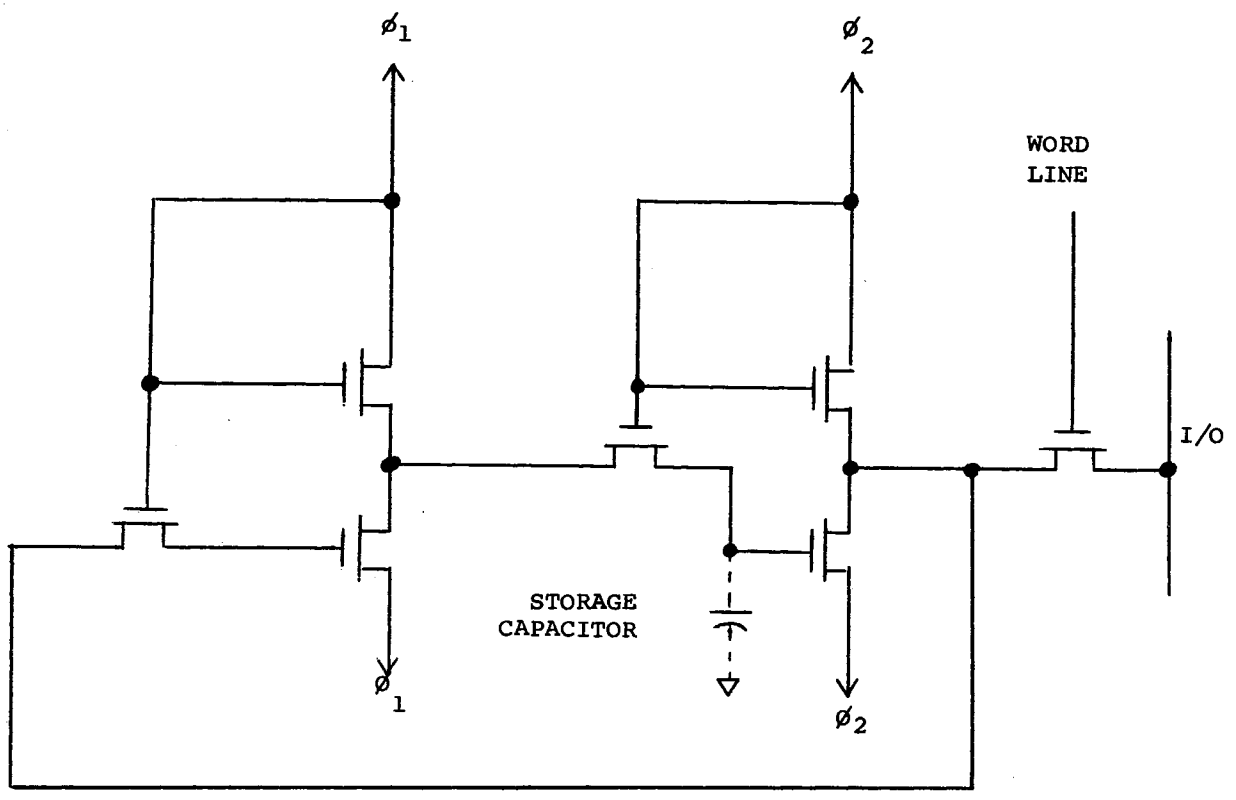


Figure 3. A Seven Transistor Minimum Size Dynamic Cell

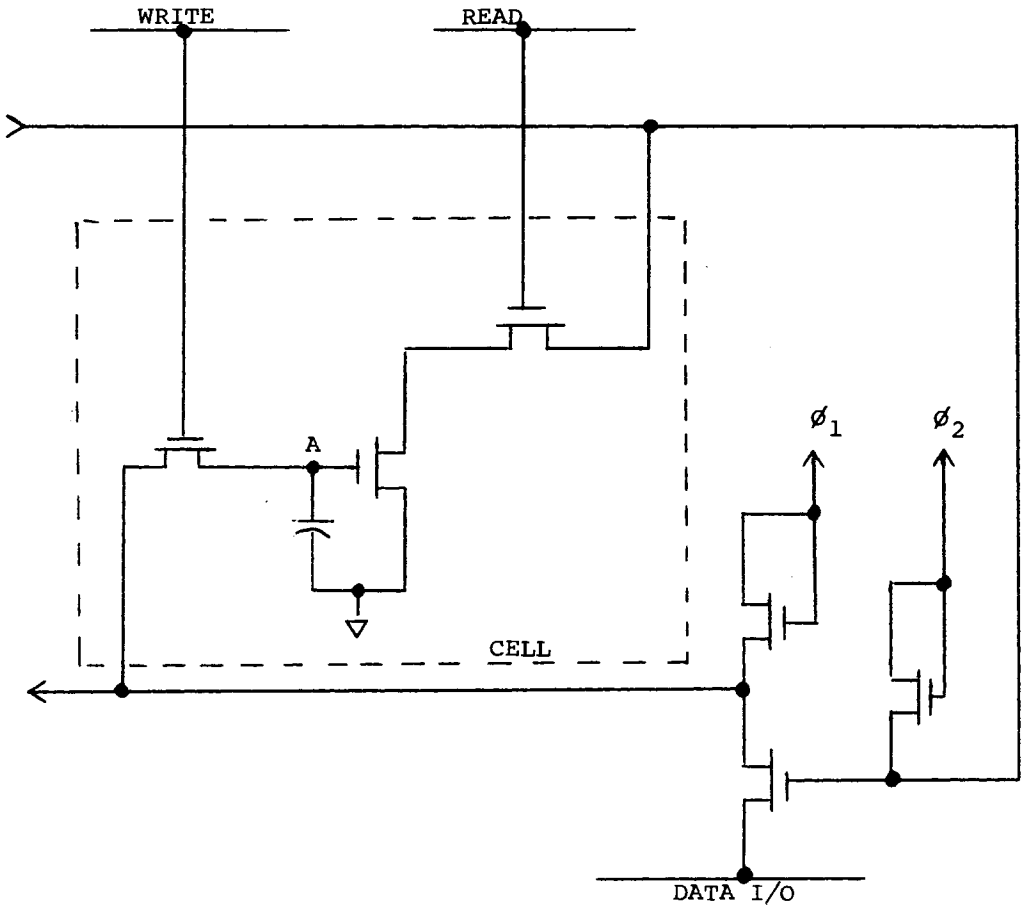


Figure 4. A Three Transistor Storage Cell With Refresh

silicon cell area required compared with the three transistor cell. Cells larger than three transistors require overall I.C. die area larger than present day processing techniques permit.

The static MOS and bipolar Read/Write memories employ the common flip-flop. Several types of static storage elements have been proposed; however, all of these elements work on the flip-flop principle.

The simplest static cell is constructed as shown in Figure 5. The cell employs the basic flip-flop and requires a number of transistors. The advantages of this cell are its noise immunity and wide electrical operating range. The cell requires no refresh operation. The major disadvantage of this cell is the large amount of silicon area and the large amount of power required. Power dissipation of the memory package is the power of the individual cell multiplied by the number of cells in the memory array. This can be an enormous amount of power when arrays are interconnected to form a memory system.

The following chapters deal only with the dynamic Read/Write memory when implementing the alterable portion of the memory subsystem. Static memories are generally not found in small machine applications because of their high cost and small storage capacity caused by the large cell size and amount of power dissipation.

The design of the memory subsystem begins with the definition of the problem. As in any design, the problem must first be defined before a solution can be found. Chapter II defines an example processor requirement. Chapters III, IV, and V outline a solution for the example while also detailing other possible approaches. In each case all steps are directed towards a worst case production design.

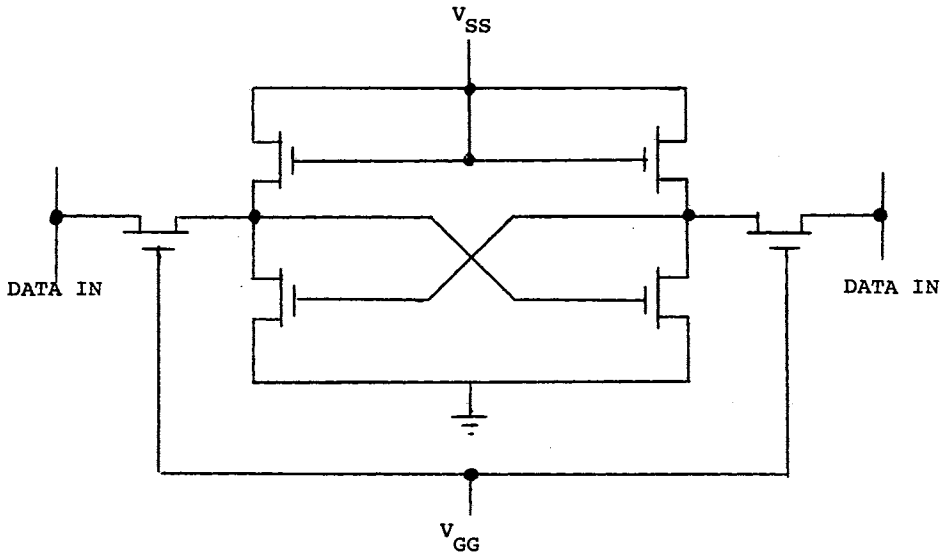


Figure 5. A Static Flip-Flop MOS Storage Cell

CHAPTER II

PROBLEM DEFINITION

Defining the Small Machine Application

Introduction

The first step in designing a memory is to define the operating conditions. The first half of Chapter II defines the conditions of the large scale integration (LSI) processor. The second half outlines and surveys the Read/Write memory IC's available in industry.

Defining the Processor

A small processor can be defined as being limited in word length, a limited number of transfer registers and instructions, and a memory structure not exceeding 100,000 words. To insure that the descriptions contained here are practical, a production design is used as a model. The design centers around a memory requirement for a Honeywell Information Systems, LSI processor. The processor is MOS p-channel, silicon gate technology.

Any processor can be described in four basic blocks, arithmetic, control, I/O, and memory. Such is the case of this example processor. Figure 6 represents the general case for a processor and will be used as a basis for defining the memory design.

The guidelines contained in the discussion of this paper are

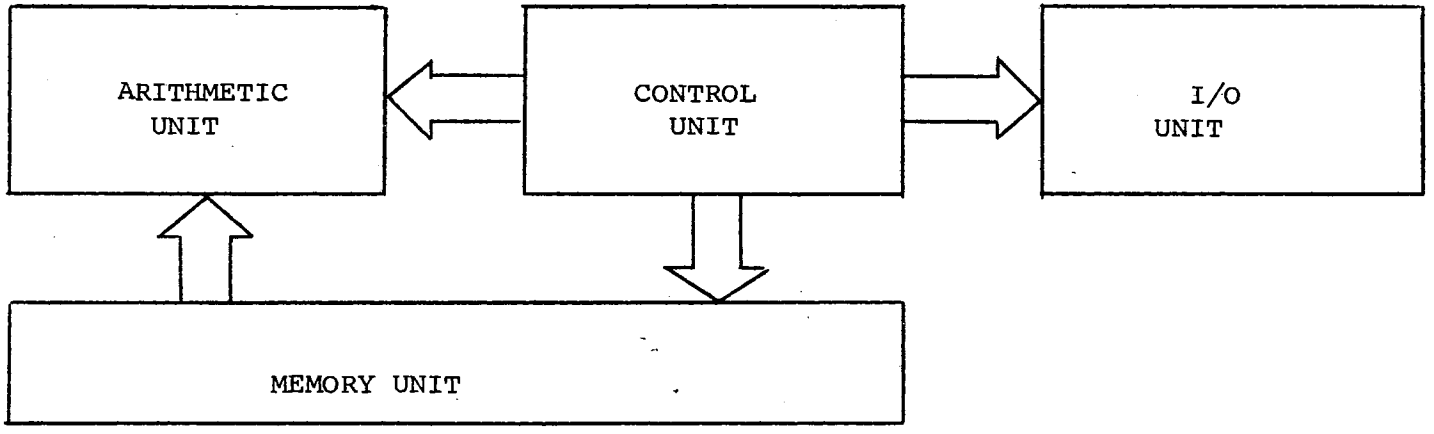


Figure 6. A Block Diagram of a Processor

primarily directed towards the memory. Therefore, descriptions of the arithmetic, control and I/O units are omitted unless they directly affect the memory operation.

Defining the Memory to Processor Interface

For the processor used in this discussion the data path for Data In and Data Out are separate. Some small machines are often designed with a bidirectional data path. The bidirectional data path is convenient for many applications and is generally used where memory interface registers and wiring are to be minimized. A functional description of the interface is shown in Figure 7. The address register consists of a 14 bit word. The Data register, both In and Out, are 16 bit words. Parity is not included in the arithmetic or control units and therefore must be designed as part of the memory.

When the LSI processor design was conceived, the use of a dynamic Read/Write memory IC had been anticipated. Because of this, a memory refresh signal was incorporated into the Control Unit. The refresh signal consisted of a refresh command for each instruction. This incorporation simplified the refreshing requirement of the memory. Any design of a processor, particularly LSI, must consider the dynamic Read/Write memory and make the proper signals available before committing the design. Reworking a system made from small scale integrated circuits is not too difficult. But, reworking a LSI design is often impractical both from a cost and time to production basis.

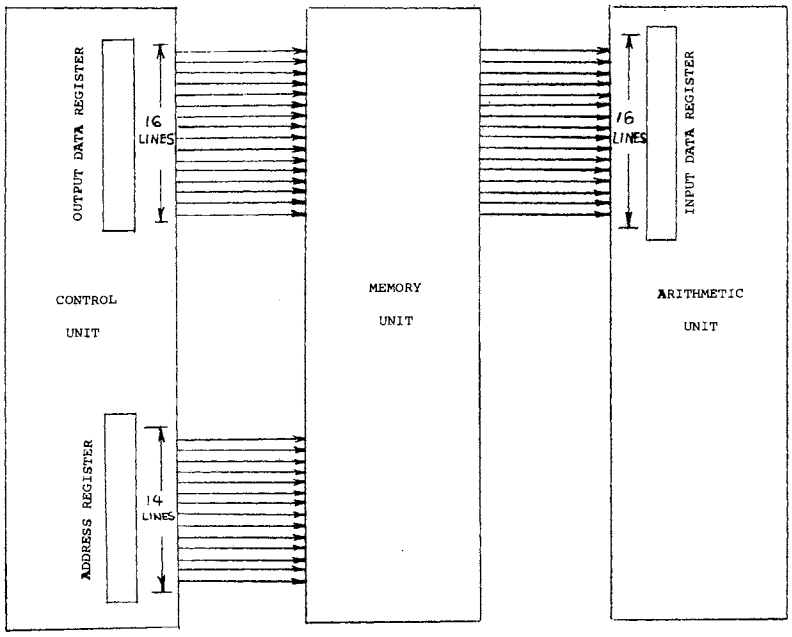


Figure 7. A Functional Block Diagram of the Processor to Memory Interface

Defining Signal Level Interfaces

One of the first steps in the design, is to insure processor to memory interface levels. Table I summarizes the processor signal levels. After the memory IC selection it will be necessary to check on processor to memory interface compatibility.

Environmental Considerations

A small processor, especially those found in numeric control and point-of-sales applications, are generally designed to an industrial environmental standard. A typical environmental specification is summarized in Table II.

An operating temperature of +65 degrees centigrade approaches the design limits of all industrial grade IC's. It is advisable to always insist on IC's being tested on a component level to their maximum temperature specification. Many designers have discovered that what works in the laboratory does not necessarily guarantee operation at all extremes of temperature in the field. Caution must be particularly exercised in a monostable timing development, such as is described later in Chapter III, for temperature causes the greatest percentage of change in timing pulse widths.

Defining the Memory Architecture

The architecture of the memory is generally described by the application of the processor. For this example, a Read/Write random access memory of 6144 words is assumed to be a necessary application requirement. This is consistent with the processor description given in the beginning of this chapter. The processor has 14 address bits which

TABLE I

PROCESSOR TO MEMORY INTERFACE SIGNAL LEVELS

D.C. Levels	Logic 0	Logic 1	Compatibility
Address Bus	2.4V	0.4V	TTL
*Output Data Bus	V_{SS} + 1.0V	0.4V	MOS
Input Data Bus	2.4V	0.4V	TTL
Clocks Refresh, Write	2.4V	0.4V	TTL

TABLE II

SUMMARIZING THE MEMORY ENVIRONMEMTAL FACTORS

Maximum Processors Outside Air Temperature	+110°F/44.4°C
Outside air to inside case temperature rise (Assumes moving air and power supply packaged with processor)	+ 30°F/15.6°C
Maximum memory operating temperature	+140°F/60°C
Allowable tolerance for printed circuit hot spots due to formation of air pockets in moving air.	5°C
Maximum Memory Operating Temperature	+ 65°C

restrict memory to 16,384 words without resorting to memory expansion techniques.

In order to be further consistent with the example processor, the Read/Write and ROM both will share a common address and data bus. Figure 8 is a pictorial structure of the memory/processor configuration.

Defining the Power Supply

The ideal memory is always power supply compatible with the rest of the machine. However, seldom is this luxury possible with a semiconductor memory. Scanning the surveys of ROM and Read/Write memories will show that few semiconductor memories are power supply compatible. Table III lists the processors and the MF 7112 power needs. Processors current requirements are not included as they do not pertain to this discussion. It is immediately obvious that the Read/Write memory is not power supply compatible. The ROM has not been chosen yet (see Chapter V) and the power supply usage was therefore not included. The designer must insure that the interface signals are truly compatible. In this case the power supply summary indicates that signal compatibility may be misleading as described by the manufacturer. Such is not the case for the MF 7112, but some manufacturers do resort to misleading specifications that can lead to many problems for the designer that does not insure input and output compatibility.

Defining Memory Interface Timing

The last definition is processor to memory timing. Often the timing between processor and the memory elements are not compatible and interface logic is required. Chapter III describes this interface timing.

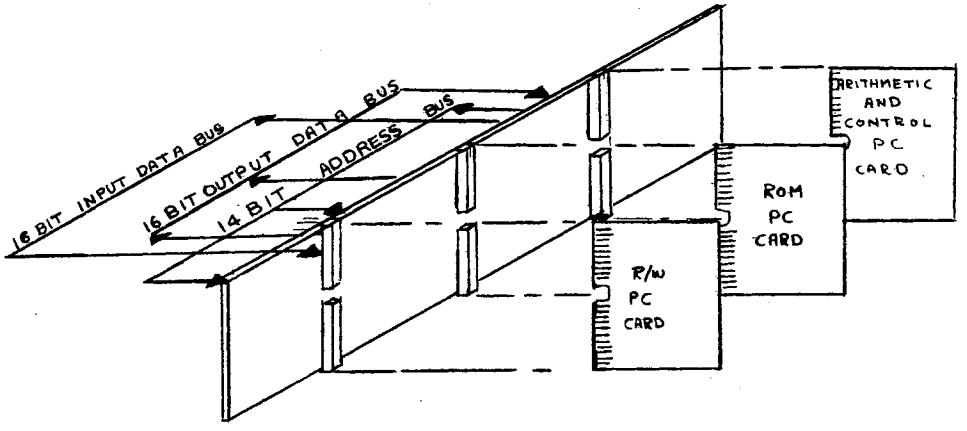


Figure 8. A Pictorial Representation of the Processor Showing Read/Write and Read Only Memory

TABLE III

SUMMARIZING THE SYSTEM POWER SUPPLY REQUIREMENTS

WHERE USED	VOLTAGES REQUIRED				
	+5.0V	+9.0V	+15.0V	-2.0V	-15.0V
Arithmetic and Control Units	✔		✔		✔
Interface Logic	✔	✔			
MF 7112		✔		✔	
I/O	✔		✔		✔

First the designer must define the processor's requirements and capabilities, and then define the memory element (in this case the MF 7112) requirements and capabilities. The best method for an overall view is by timing diagrams. Figure 9 is the processor's Read Cycle; Figure 10 is the processor's Write Cycle; Figure 11 is the MF 7112 Read and Write timing.

Surveying the Semiconductor Read/Write Memory

Introduction

The selection of a memory element is usually a function of the systems performance requirements and the memory cost goals. Many choices are available to the designer. This section surveys the present day market of semiconductor Random Access Read/Write Memories, the technologies, performance and cost are used as guidelines in graphing the available market.

The designer is presented with a maze of choices when he first begins his memory selection. Armed with his cost goals or the performance goals as primary factors, he finds that these two parameters are both directly related to the semiconductor technology. The axiom of circuit design used from the beginning of electronics still holds true for semiconductor memories. This axiom, the faster you must operate the more it is going to cost, often limits the designers choice. In general, the bipolar technology is chosen for applications with cycle times less than 100 nanoseconds, while the MOS technology is used for cycle times greater than 100 nanoseconds.

Another rule of thumb for design cost considerations is, that bipolar memories are two to four times the cost of MOS memories. On the

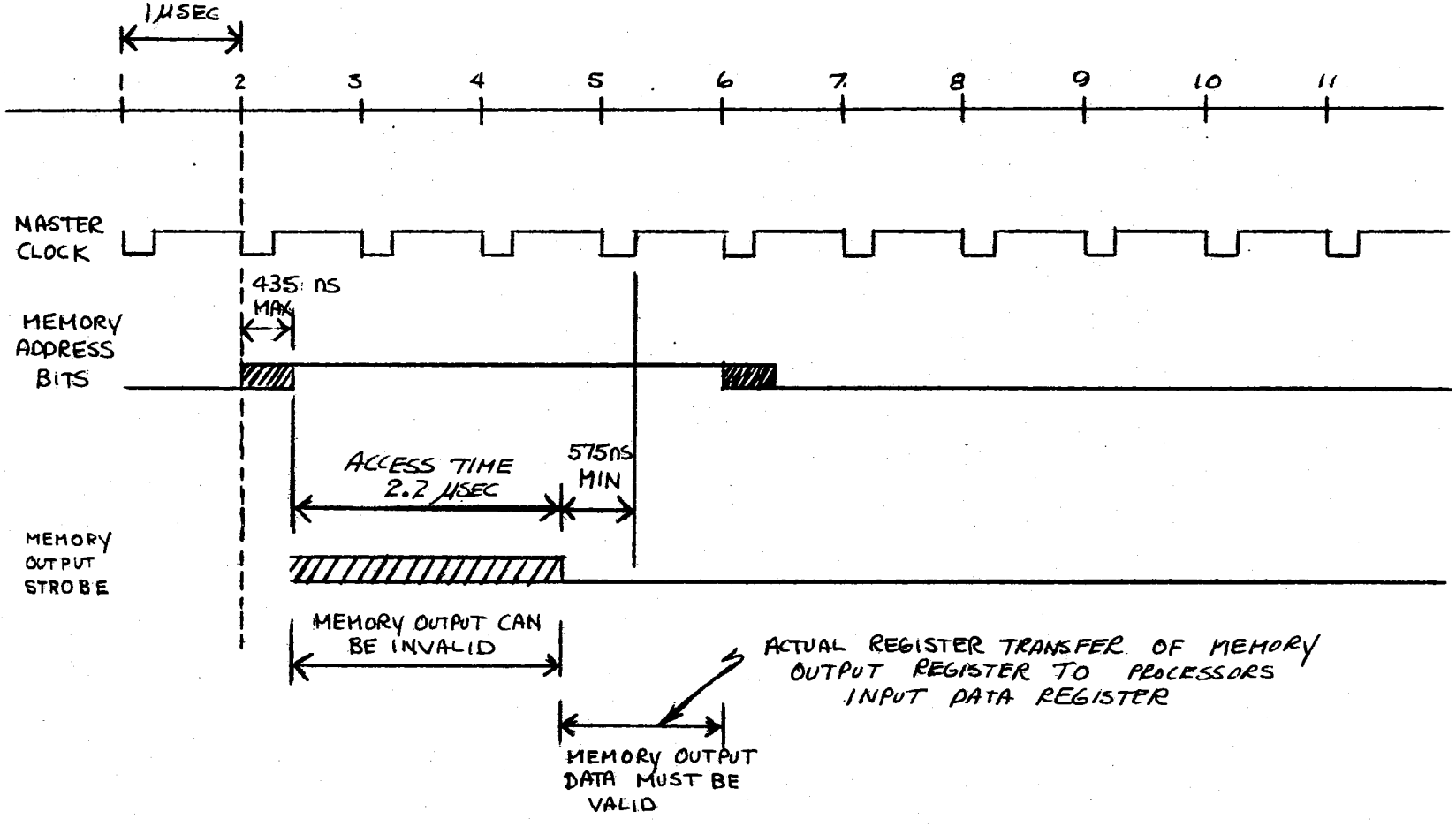


Figure 9. Timing Diagram of a Typical Processor Read Cycle Including Worst Case Design Parameters

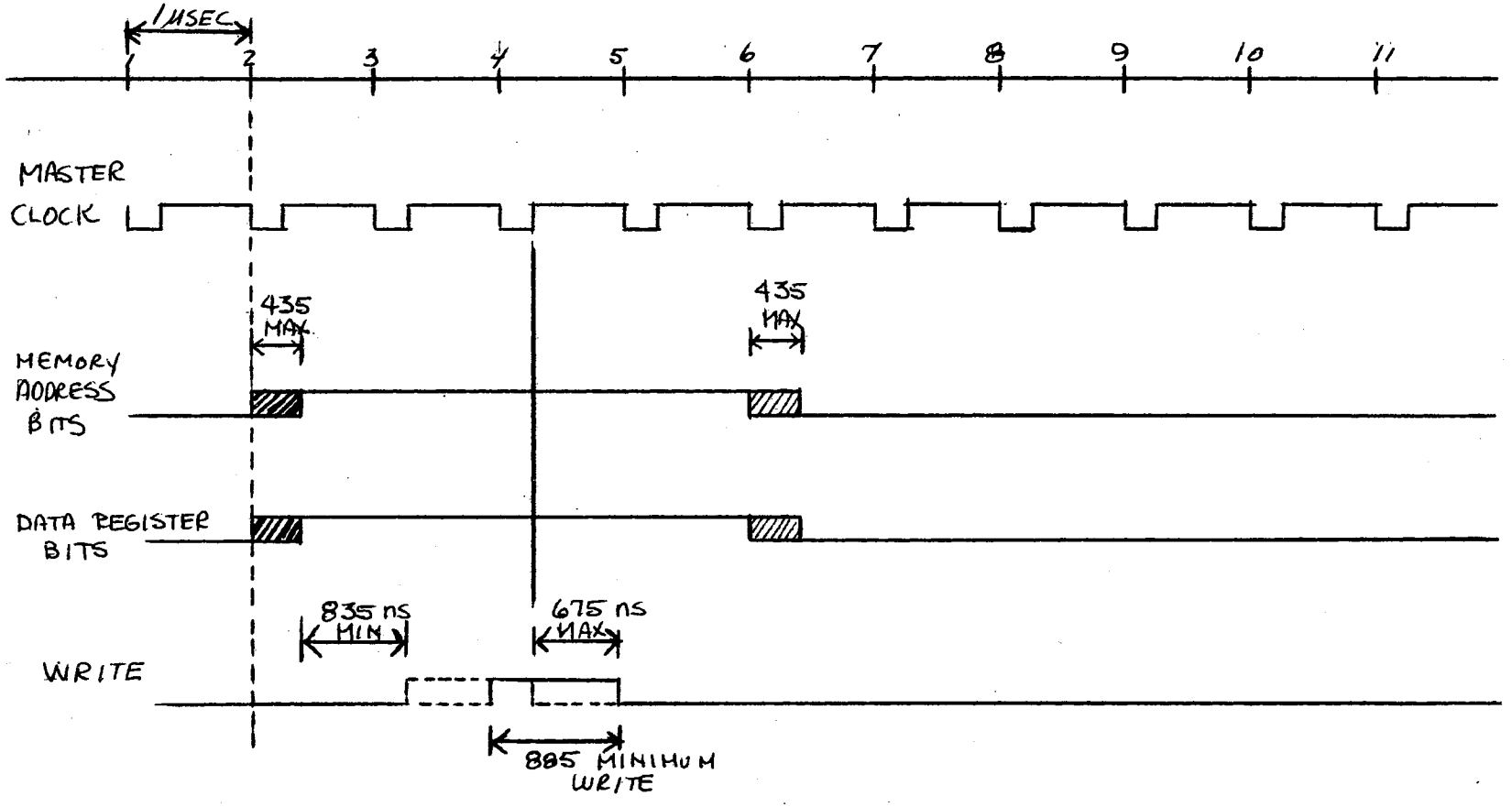


Figure 10. Timing Diagram of a Typical Processor Write Cycle Including Worst Case Design Parameters

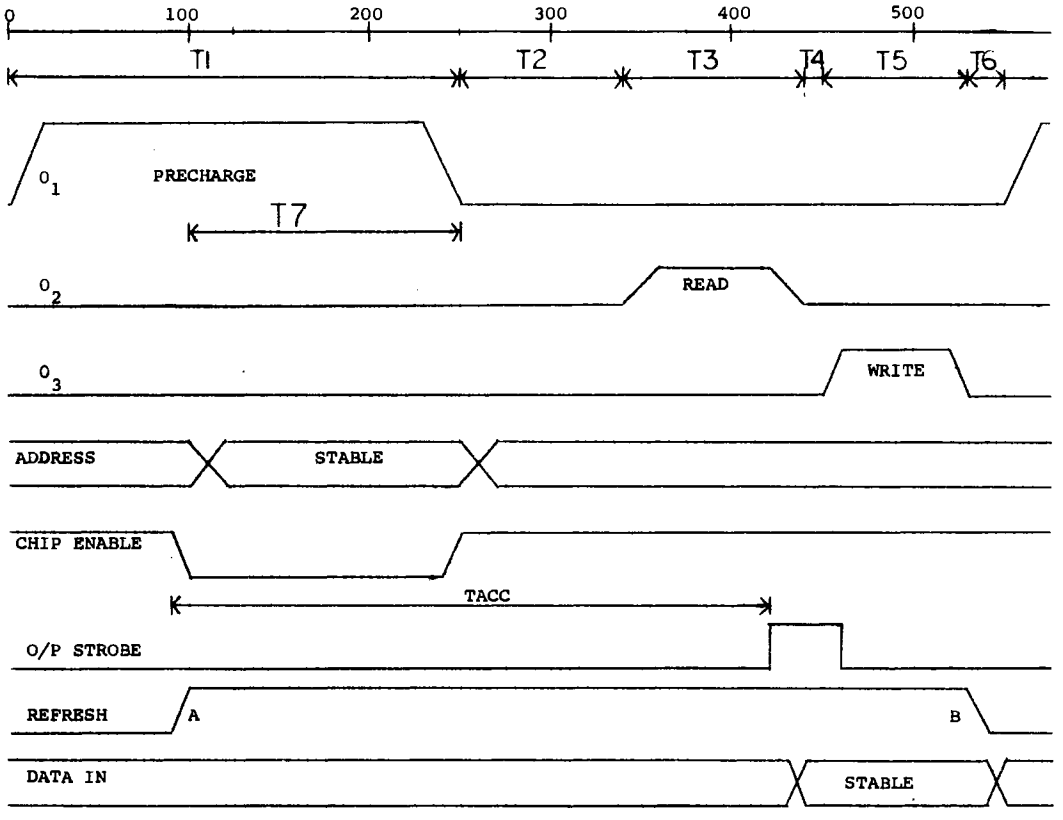


Figure 11. The MF 7112 Dynamic MOS Read/Write Memory Read and Write Cycle Timing Requirements

other hand, bipolar memories are simpler to use and require less interface circuitry.

For small machine applications the designer must compare both technologies for the best system cost/performance trade off. Table IV lists some of the more important items to be used in selecting a memory for small machine applications. Their order of importance is also listed even though these are variable depending on the application. Notice that some items must be treated with an equal importance factor.

Industry Survey

Both the MOS technology and the bipolar technology can be divided into various branches, with each branch a cost/performance trade-off. Out of necessity, the MOS technology must be sub-divided into the majority carrier channel designation, i.e., p-channel and n-channel. The bipolar technology can be divided into the type of transistor interconnections used within the memory element. Figure 12 represents the structure of the bipolar and MOS memories, and shows the choices available to a designer. The bipolar branches are straight forward with two choices presently available, TTL and ECL. On the other hand, when the access and cycle time of a memory allow MOS to be the chosen technology, there are many branches available. The MOS memories are broken down into the three common technologies and then into the majority carrier channel designator. Also, added are the complementary MOS (C-MOS) and the silicon on sapphire (SOS) technologies.

Shown in Figure 13 is a generalized curve representing the technology trends of the semiconductor memory usage. It is important that the designer consider the market trends for his product during the

TABLE IV

MEMORY SELECTION CONSIDERATIONS

IMPORTANCE WEIGHING FACTOR	SELECTION CRITERIA
1	CYCLE AND ACCESS TIME
1	MEMORY ELEMENT (IC) SIZE
1	COST
2	EASE IN USAGE
2	PHYSICAL PACKAGING REQUIREMENTS
2	TEMPERATURE EXTREMES
3	POWER SUPPLY COMPATIBILITY
3	AVAILABLE SOURCES
4	SIGNAL LEVEL COMPATIBILITY
4	SPECIAL ADDITIONAL CIRCUITRY
5	MEMORY ELEMENT (IC) ORGANIZATION
6	STATIC OR DYNAMIC OPERATION
7	INCOMING INSPECTION, INITIAL TEST DIFFICULTY
7	SPECIAL HANDLING REQUIRED
8	SPECIAL PRINTED CIRCUIT CARD RESTRICTIONS

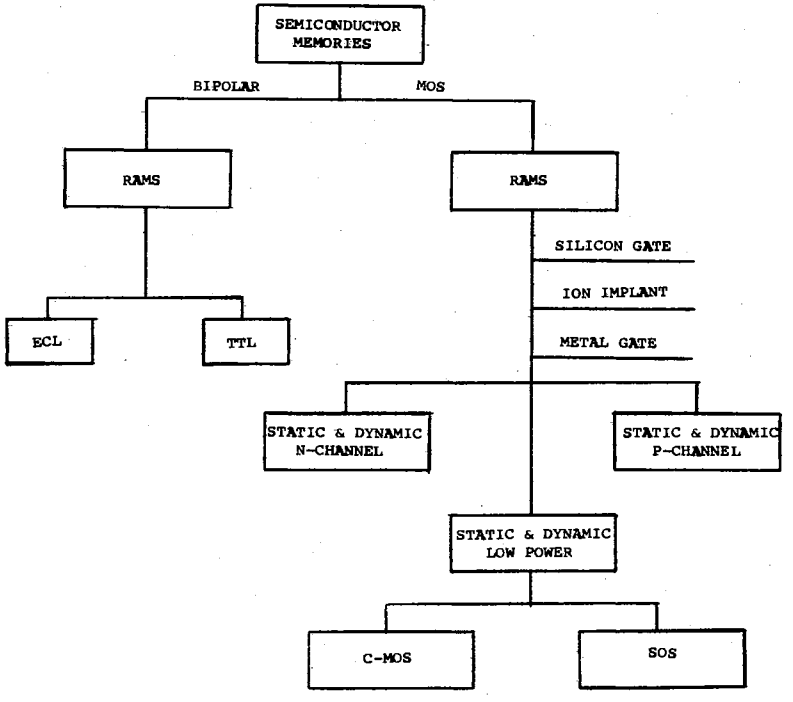


Figure 12. The Choices of Semiconductor Read/Write Random Access Memories

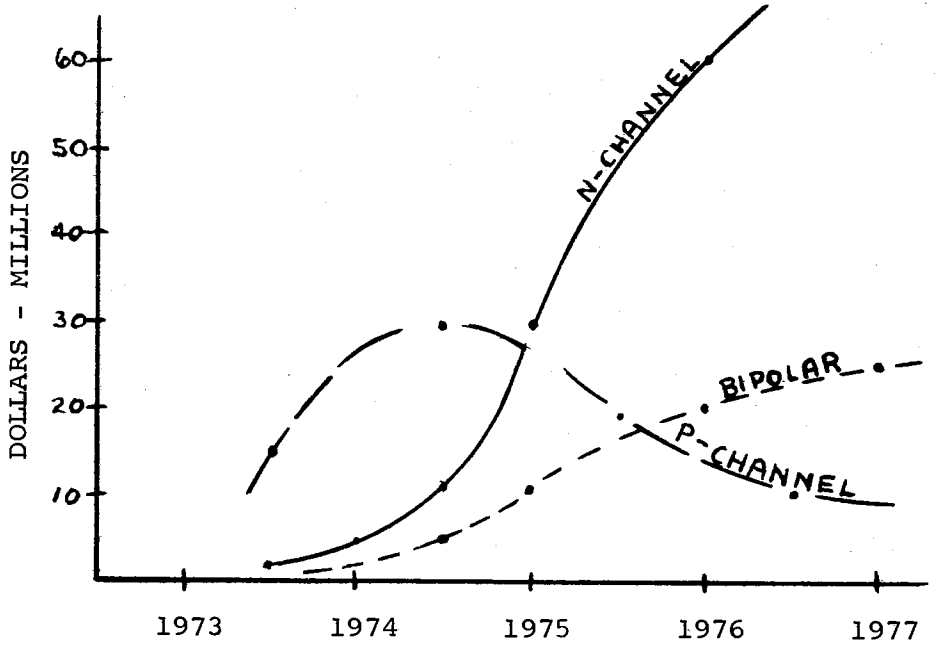


Figure 13. A Generalized Curve Showing The Trends of the Memory Technology

product's prime life period. For instance, the curves show that a p-channel Read/Write memory would be the best choice for a product whose life was to end in the 1974 to 1975 period. On the other hand, for products whose life expectancy exceeded 1974 the choice of an n-channel Read/Write memory would best suit the cost trade-off of the product.

The trends shown in Figure 13 are based on personal research gathered during interviews with various semiconductor memory manufacturers.

The three tables, Table V, Table VI, and Table VII represent semiconductor memory technology divided into their two major components, bipolar and MOS. Each chart includes the Random Access Read/Write Memories presently available and forecasted for announcement prior to June, 1974. The bipolar memories are further sub-divided into TTL (Table VI) and ECL (Table VII).

TABLE V

A TABLE OF TTL READ/WRITE SEMICONDUCTOR RANDOM ACCESS MEMORIES

POPULAR INDUSTRY NUMERICAL DESIGNATION	MEMORY ORGANIZATION	MEMORY ELEMENT SIZE IN BITS	PROCESS TECHNOLOGY	CYCLE TIME IN ns	MANUFACTURER
3101	16 X 4	64	TTL	80	Fairchild, Monolythic Memories, Intersil, National Semiconductor, †Intel , Signetics, Motorola, Texas Instruments
5503	256 X 1	256	TTL	85	Fairchild, National Semiconductor, †Intersil , Signetics, Intel, Texas Instruments
93415	1024 X 1	1024	TTL	100	†Fairchild , Intersil

† Indicates manufacturer whose numerical designator is listed.

TABLE VI

A TABLE OF ECL READ/WRITE SEMICONDUCTOR RANDOM ACCESS MEMORIES

POPULAR INDUSTRY NUMERICAL DESIGNATION	MEMORY ORGANIZATION	MEMORY ELEMENT SIZE IN BITS	PROCESS TECHNOLOGY	CYCLE TIME IN ns	MANUFACTURER
10145	16 X 4	64	ECL	15	† Motorola , Texas Instruments
10148	64 X 1	64	ECL	15	† Motorola , Texas Instruments
10147	128 X 1	128	ECL	15	† Motorola
10144*	256 X 1	256	ECL	25	† Motorola , Texas Instruments
10150*	256 X 4	1024	ECL	25	† Motorola
10152*	256 X 1	256	ECL	15	† Motorola
10405	128 X 1	128	ECL	15	Fairchild ,†Texas Instruments

* Indicates 1974 Production

† Indicates Manufacturer whose numerical designator is listed.

TABLE VII

A TABLE OF MOS READ/WRITE SEMICONDUCTOR RANDOM ACCESS MEMORIES

POPULAR INDUSTRY NUMERICAL DESIGNATION	MEMORY ORGANIZATION	MEMORY ELEMENT SIGN IN BITS	PROCESS TECHNOLOGY (CHANNEL)	CYCLE TIME ns	MEMORY CHARACTERISTICS	TTL COMPATIBLE	MANUFACTURER
1101	256 X 1	256	p	1000	Static	Yes	†Intel, MIL, National Signetics, General Instruments, Mostek
1103	1024 X 1	1024	p	560	Dynamic	No	†Intel, Signetics, MIL General Instruments, AMI*
MM5260	1024 X 1	1024	p	680	Dynamic	Partial	†National , Plursey
MK4006P	1024 X 1	1024	p	680	Dynamic	Partial	†Mostek , AMI
AMS6001	1024 X 1	1024	p	480	Dynamic	No	†AMS , Texas Instrument
AMS6003	2048 X 1	2048	p	500	Dynamic	No	†AMS , National, Motorola, Signetics
2105	1024 X 1	1024	n	200 (95 access)	Dynamic	Partial	†Intel
7001	1024 X 1	1024	n	200 (100 access)	Pseudo-static	Partial	†AMS , Motorola
S6605	4096 X 1	4096	n		Dynamic	Partial	†AMI , Motorola
2107	4096 X 1	4096	n		Dynamic	Yes	†Intel
MF7112	4096 X 1	4096	n	680	Dynamic	Partial	†MIL
TMS4030	4096 X 1	4096	n	500	Dynamic	Yes	†Texas Instrument
MK4096	4096 X 1	4096	n	400	Dynamic	Yes	†Mostek

†Indicates Manufacturer whose numerical designation is listed.

CHAPTER III

METHODS OF DEVELOPING THE MEMORY INTERFACE LOGIC

Introduction

Chapter Summary

The major portion of the text is developed in this chapter. Chapter III considers the adaptation of the memory timing requirements to those of the processors. Discussed are two methods of interfacing the processors master clock, address, and data registers to those of the memory. Also developed, are the dynamic memory (MF 7112) refresh interface requirements considering the four possible operating modes of a processor. The final interface network considered is the generation of the WRITE strobe in relation to the processors and memory timing.

Types of Interfaces

The timing between the control unit and the memory can be either asynchronous or synchronous. The memory interface logic must be designed dependent on this timing. Most processors operate synchronously as a mutually chosen time period does not have to be arrived at. For this reason, this discussion will be directed towards the synchronous operation.

There are many ways to develop the interface logic timing. Much depends on the operation of the control unit and the needs of the memory

storage IC. The interface logic can be divided into three separate sections, each related and dependent on the other for operation. These sections are the: (1) timing, development of memory clock pulses; (2) the dynamic memories refresh logic; and (3) the WRITE strobe synchronization logic.

Timing Methods

For a static Read/Write memory IC, interface logic is not required unless it is operated asynchronously.

For a dynamic Read/Write memory IC, the adaptation to a processor is much more complex, requiring the interface logic described in the preceding paragraph.

The most logical place to begin the design of the memory interface logic is in the timing area. There are two methods of developing the clock pulses that are practical and economical for industrial use. The first method is the familiar ring counter and counter decode system. The second method is the more economical of the two but not as reliable or accurate. This is the monostable multivibrator method. In this method the monostable multivibrators are series one after the other to generate memory clock pulses using signals and master clock from the Control and Arithmatic units.

The ring counter is used because of its accuracy and low cost. Pulse generation is done with a minimum of material; however, in order to have logical resolution, a master clock must be provided that, by rule of thumb, has a frequency that is eight (8) times the memories cycle time. This restriction makes the ring counter unsuitable for some applications such as battery operation after power is turned off.

On the other hand, the series multivibrator is often used in applications where the ring counter method is found to be undesirable. The multivibrator can operate from a low frequency processor master clock and still generate all of the necessary timing pulses and strobes. However, the multivibrator method can not operate at frequencies above five (5) megahertz in this type of application because of the extremely large minimum to maximum pulse width shifts caused by susceptibility to environmental, and power supply changes. Also, the multivibrator is susceptible to large variations from one device to another. The multivibrator also has the worst susceptibility to noise transients of the two methods.

Both methods are presented in the following sections and must be chosen depending on the application and memory storage IC. For these examples, the MF 7112 Read/Write random access memory is used. The timing is developed in accordance to the problem definition of Chapter II.

The Ring Counter

Choice of the Logic Clock Period

Examining the timing diagram of Chapter II shows that the system requirements for both a Read Cycle and a Write Cycle is approximately 2.2 microseconds. The Read/Write IC itself is capable of a 490 nanosecond read cycle and a 680 nanosecond write cycle. Further, the basic clock frequency of the system is one (1) megahertz. The first decision to be made is deciding what frequency to operate the memory cycle.

The requirement is to make a compromise between memory performance and system requirements, comparing the system clock frequency with the

Read/Write IC memory performance. The most obvious cycle period would be one (1) megahertz. This means that the Read/Write IC will be operating at a cycle time less than its rating, and the system access and cycle time requirements are met. Figure 14, is the generalized timing diagram of the system and memory IC.

In general the system almost always dictates the memory access and cycle times. Any design would have to consider the system's requirements with the memory element's capability.

Implementing the Ring Counter Design

A block diagram of the desired function is shown in Figure 15. A crystal controlled oscillator is used to generate both the system and memory timing. A count down circuit is used to generate the one (1) megahertz system clock, while the ring counter clock comes from the 8 megahertz oscillator.

The state equation for the shift counter can be written as

$$A_n(\tau) = A_{n-1}(P) \quad . \tag{1}$$

The equation is interpreted as: the state of the n^{th} flip-flop of the register, following a clock interval, is the state of the $(n-1)^{st}$ flip-flop occurring at clock pulse P.

The register is connected with the flip flop A_8 output feeding back to the input of flip flop A_1 in the manner of a circulating register such that the input of flip flop A_1 is dependent on the output of A_8.

In order to initialize the register to the proper state during power turn-on a power-on state is connected to the parallel input of the register. The parallel input of the register is set such that all inputs

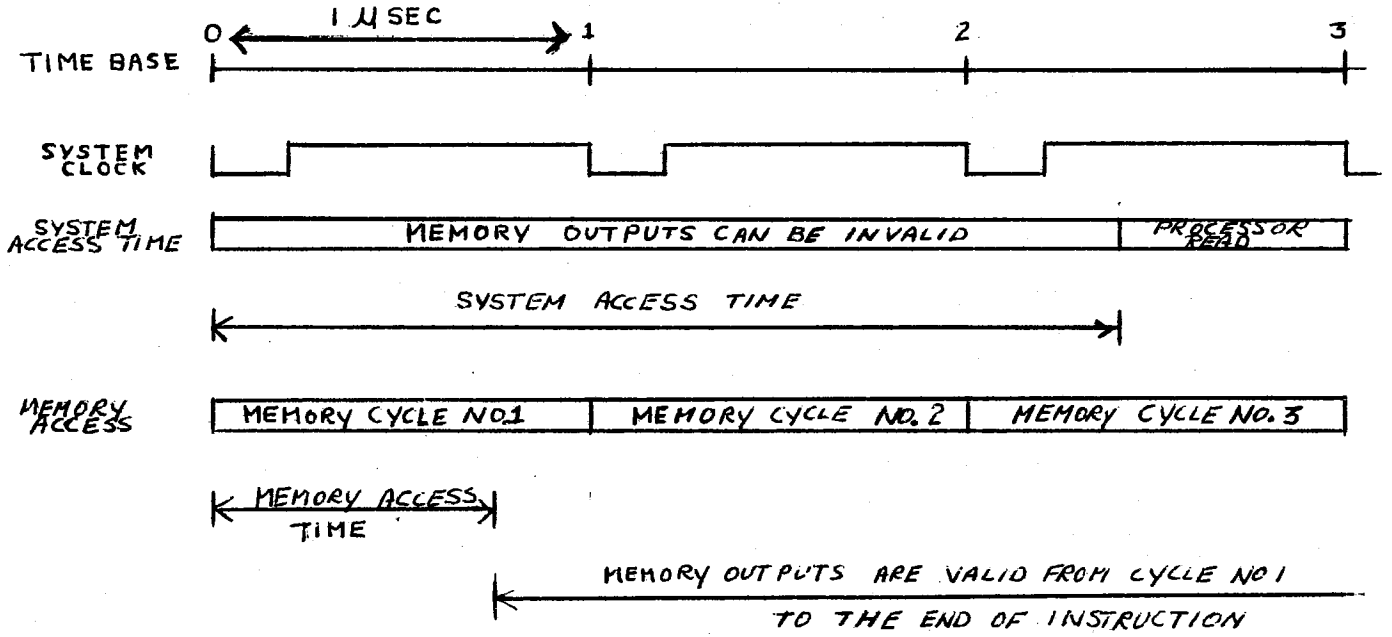


Figure 14. A Generalized Timing Diagram of the System and Memory IC

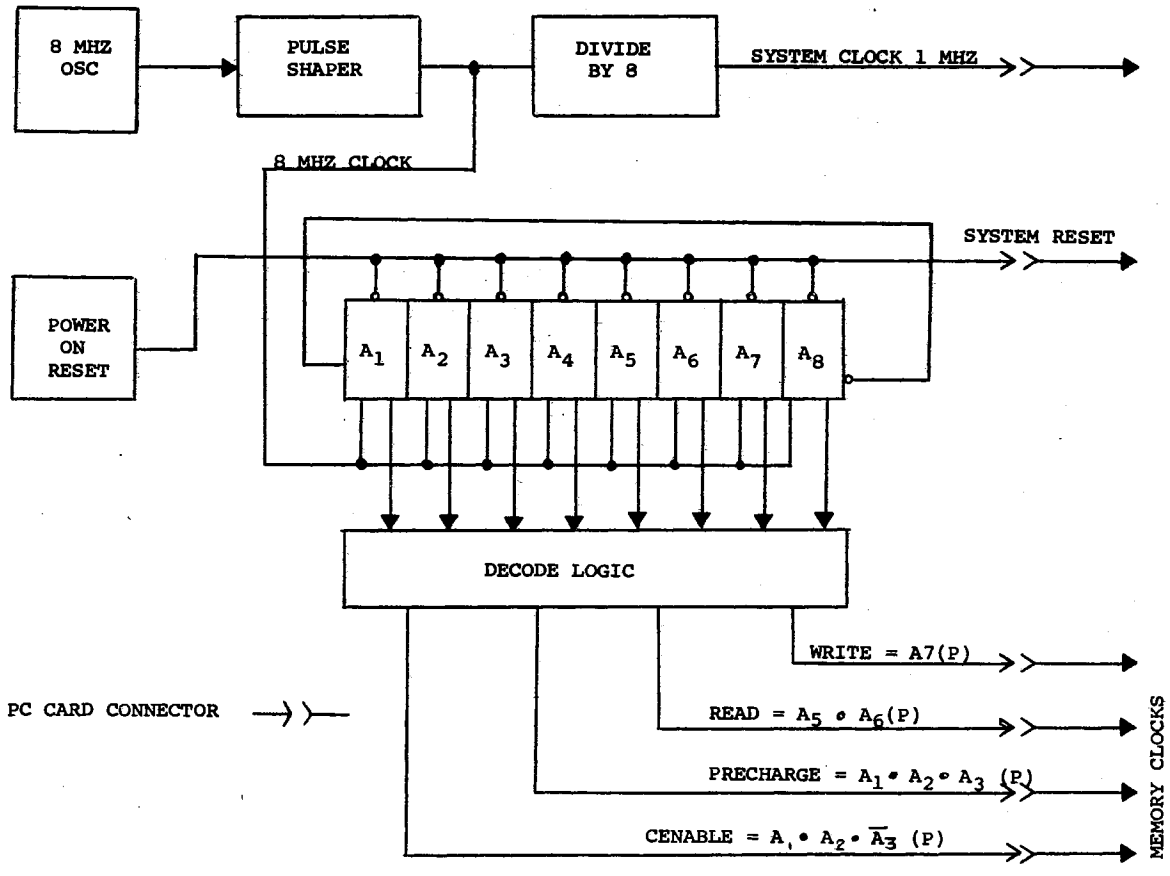


Figure 15. A Block Diagram of the Ring Counter Including Decoding

are at a ground potential except flip flop A_1. When the power-on state returns to the operating state, the inputs of flip flop A_1 are such that the start bit appears on the output of A_1 and is shifted through the register at a 8 megahertz rate until the bit appears on the output of flip flop A_8. The bit is then recirculated back to flip flop A_1 and the cycle started over again. The equation for the register is written as

$$A_n(\tau) = (A_{n-1})P + C \quad , \tag{2}$$

where C is the clearing pulse resulting from the power-on initialization.

Decoding the Clock Pulses From the Register

The three clock pulses required by the MF 7112 are derived from the register by decoding the output of the register. Examining the MF 7112 timing diagram from Chapter II, the equations for the clock pulses are written as

$$\text{Precharge} = A_1(P) \cdot A_2(P) \cdot A_3(P) \tag{3}$$

$$\text{Read} = A_5(P) \cdot A_6(P) \tag{4}$$

$$\text{Cenable} = \overline{A_1(P)} \cdot A_2(P) \cdot A_3(P) \quad . \tag{5}$$

A fourth pulse is also required. This pulse will be called, End of Cycle, and will be defined later in this chapter under the sub-title of Refresh. The pulse is defined in equation form as

$$\text{End of Cycle} = A_8(P) \quad . \tag{6}$$

The delay between the Precharge pulse and the Read pulse required by the MF 7112, is guaranteed by the clock delay of A_4.

A Write pulse is required by the MF 7112 in the logical 1 state for an alter (write into) memory to occur. The MF 7112 write pulse must occur after the Read pulse and be a minimum of 100 nanoseconds wide. The equation for the Write pulse can be expressed as

$$\text{Write} = A_7(P) \quad . \tag{7}$$

The Write pulse applied at the input of the MF 7112 is the logical AND of the system Write derived from the control unit and the register pulse $A_7(P)$. More of the operations of the Write signal operation is discussed in the latter part of this chapter.

Figure 16 is the schematic of the final implementation of the Ring Counter design. Figure 17 is the overall timing diagram of the system and the Ring Counter design.

Ring Counter Design Analysis

The design continues with an analysis of the component cost. This is used to determine the related merits of this technique when compared with the monostable technique to be taken up later in this chapter.

The address register shown in the timing diagram will not be included in the cost analysis, nor will the power-on reset, or the oscillator as all of these elements will be required in either technique. There might be some question as to why the cost of the oscillator is omitted. The assumption is, that the system must have a crystal controlled timing source regardless of the technique used. Therefore, the oscillator is going to be in the system somewhere and being used in a dual role as both system and memory master clock. The cost of the oscillator can then be eliminated as an integral part of the memory cost.

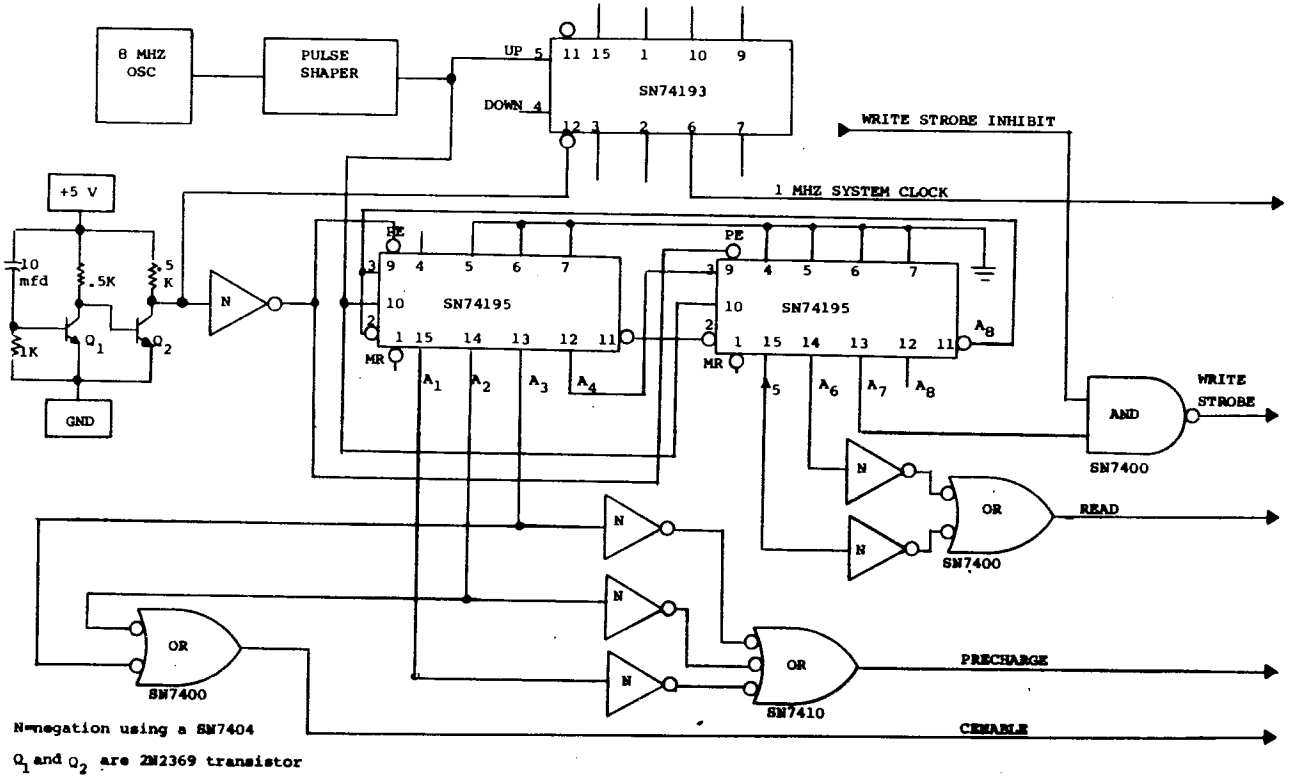


Figure 16. Final Implementation of the Ring Counter Method

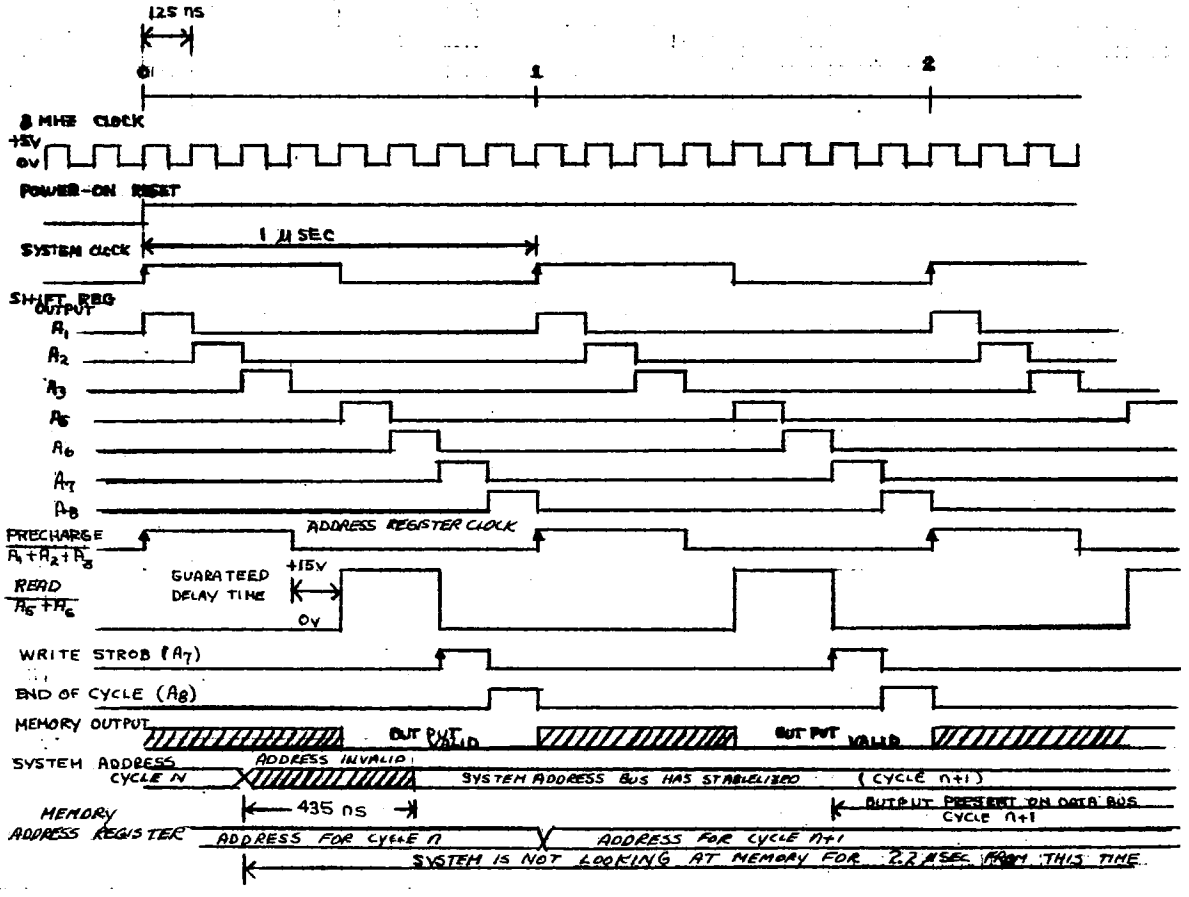


Figure 17. Overall Timing Diagram of the System and the Ring Counter Method

TABLE VIII

SUMMARIZING THE COST OF THE RING COUNTER

COMPONENT	FUNCTION	QUANTITY	COST
SN74193	COUNTER	1	$1.25
SN74195	SHIFT REGISTER	2	$1.50
SN7404	HEX INVERTER	1	$0.25
SN75361	AND GATE, HIGH VOLTAGE	2	$4.00
SN7400	2 INPUT AND GATE	1	$0.25
SN7410	3 INPUT AND GATE	1	$0.25
2N2369	TRANSISTOR	2	$0.50
RESISTOR	1/4 WATT	3	$0.15
CAPACITOR	10 MICROFARAD	1	$0.25
		TOTAL	$8.40

Timing Generation With Monostable Multivibrators

General

One of the simplest and least expensive methods of developing pulses, such as the MF 7112 memory IC requires, is to use monostable multivibrators (one-shots). However, the penality of simplicity is the danger of false triggering and large tolerance variations. A large amount of attention must be paid to the printed circuit board layout design and to power supply filtering and individual IC decoupling to avoid these undesirable effects. Monolithic monostables prove to be attractive both from a physical area and cost effectiveness. Such monostables as the single element 9601, or the dual element (two in a dual in-line package) 9602, are suitable for design. Both of these devices are similar in operation and performance. Pulse width tolerances for worst case design analysis should be considered as $\pm$ 25 per cent. This accounts for individual device to device substitution, 5 per cent power supply variations, and temperature variations of 0 degree centigrade to 70 degree centigrade.

Both the 9601 and the 9602 multivibrators are edge triggering devices and are therefore particularly subject to false triggering from such phenomena as inductive input signal ringing and power supply noise. A great deal of care in printed circuit layout and component placement must be exercised to prevent undesirable input triggering. Rise and fall times of input signals must be controlled to prevent the monostable from assuming the characteristics of a high gain amplifier with a 180 degree phase shift. This can occur when a TTL device is transitioning in the active threshold region lying between a logical "1" and "0". With a slow

rise or fall time (1 millisecond or greater) from the preceding driving stage, the monostable could act as an amplifier. With a long enough time period for the input transition, the monostable will begin to oscillate.

Considering Refresh

Using the previous criteria that a refresh cycle amounts to a memory "Read" cycle with an invalid output, makes the design of the memory timing requirement somewhat simpler. The only requirement of the refresh initialization pulse is that the series of monostables are properly triggered such that a normal Read memory cycle occurs when refresh is commanded. More on refresh will be discussed later in this section after the memory clocks are developed.

Clock Pulses

The same memory timing for a Read and Write cycle used in the shift register ring counter design is valid when using the monostable. Repeating these conditions in equation form again for convenience,

$$\text{Precharge} = A_1(P) \cdot A_2(P) \cdot A_3(P) \geq 300 \text{ ns} \tag{8}$$

$$\text{Read} = A_5(P) \cdot A_6(P) \geq 180 \text{ ns} \tag{9}$$

$$\text{Write} = A_7(P) \tag{10}$$

$$\text{Cenable} = A_1(P) \cdot A_2(P) \cdot A_3(P) \geq 200 \text{ ns} \tag{11}$$

The monostable design now produces the pulses previously accounted for by decoding the ring counter.

The equations for pulse width requirements using monostables can be rewritten as

$$\text{Precharge} = t_1 \ (PW_1) \tag{12}$$

$$\text{Read} = t_3 \ (PW_2) \tag{13}$$

$$\text{Write} = t_4 \ (PW_3) \tag{14}$$

$$\text{Cenable} = t_5 \ (PW_4) \tag{15}$$

Defining PW as pulse width, and t_n as time one, two, etc., the equations read as

Precharge is equal to pulse width 1 at time t_1 of the shift register, ring counter timing.

Address Register

Referring to Figure 18, an address register pulse is caused to occur at the initial edge of the precharge pulse. This timing must now, because of the monostable approach, be accomplished differently than the ring counter to insure that the address does not change during the precharge clock. It is also essential that the address change on the memory subsystem level occurs synchronously with the other memory timing. Observing the timing diagram of the memory Read cycle (Chapter II) it will be noted that the LSI processor used as the example model, has a minimum address register true time to the address bus of 15 nanoseconds and a maximum of 685 nanoseconds. The maximum propagation delay of the address register is the cause for the address register on the memory sub-system level. More on the implementation of the memory address register will be discussed later in this section.

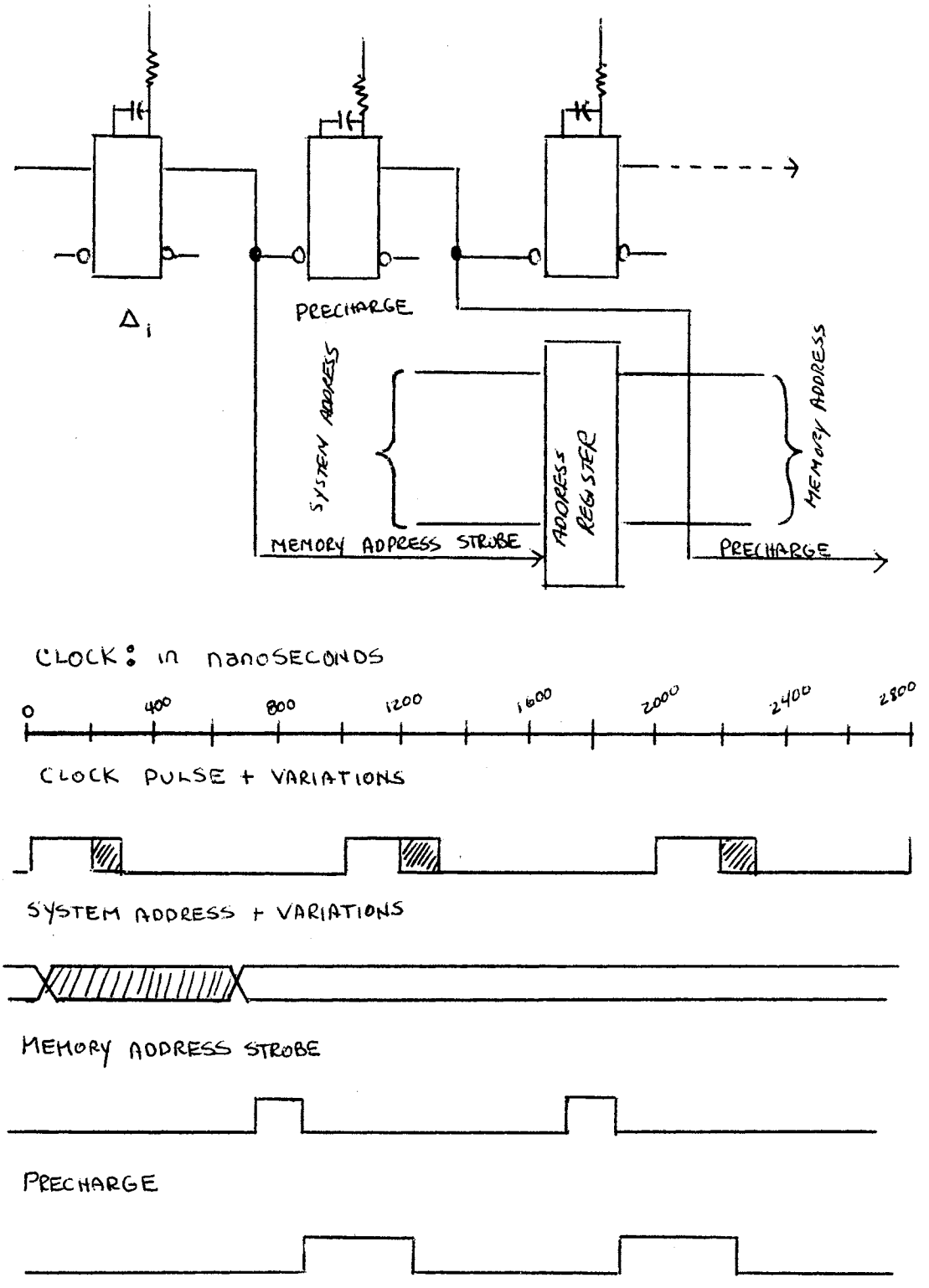


Figure 18. Timing Diagram and Logic Diagram of the Address Register Precharge Relationship

Desired Timing

Before proceeding further, the sequence of events on a time basis needs to be defined. For an instruction that commands memory fetch (Read) a Read Cycle is desired. For an instruction that commands a memory alter, a Write Cycle is desired. The combination of processor central clock, change of address, and data register changes in conjunction with memory timing requirements are shown in Figure 19. This figure summarizes the timing relationships between system and memory sub-system. Each of the one megahertz increments of the master central clock must be examined. Figure 20 represents an example of one clock period and what transpires in that period.

Two timing points that require special attention are at time t_0 and t_f of Figure 20. This is when the address change occurs for any instruction. Care must be taken that the address from the control unit does not change in the time frame of the precharge pulse. This is an additional requirement from those described under Address Register. Once again, the addition of the memory sub-system address register will prevent any undesirable address changes due to element propagation times.

Additional care must also be exercised when the refresh address is switched into the memory address bus. This is discussed in this chapter under Refresh.

Implementing the Logic Equations

The basic series of monostable logic can now be implemented using Equations 12 through 15. Shown in Figure 21 is the implementation of Equations 12 through 15 using series monostable multivibrators.

The timing pulse and delay generation are implemented using seven

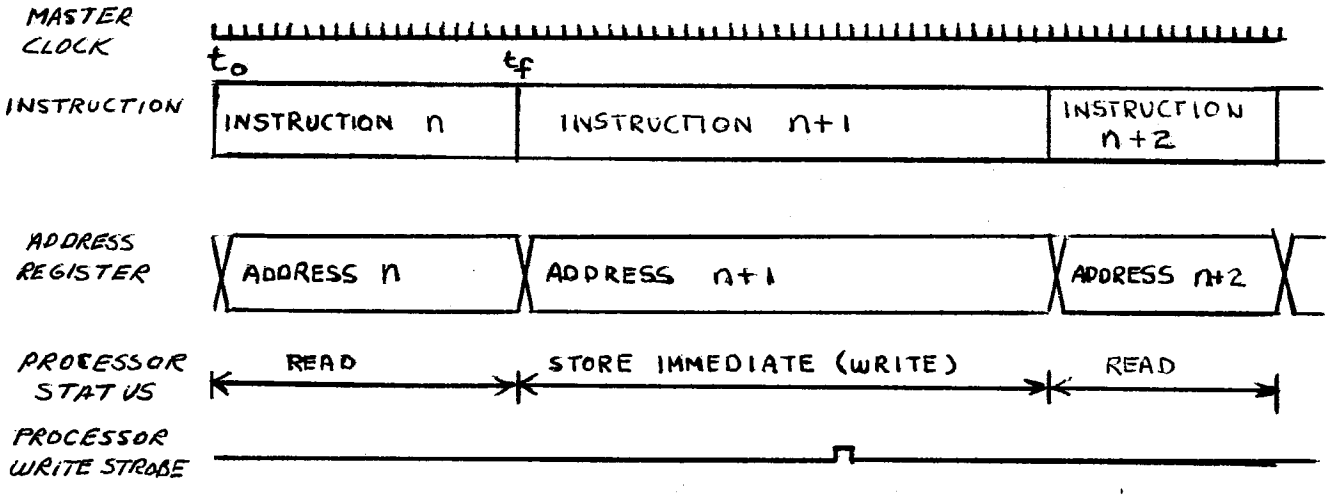


Figure 19. Processor Instruction Level Timing Diagram

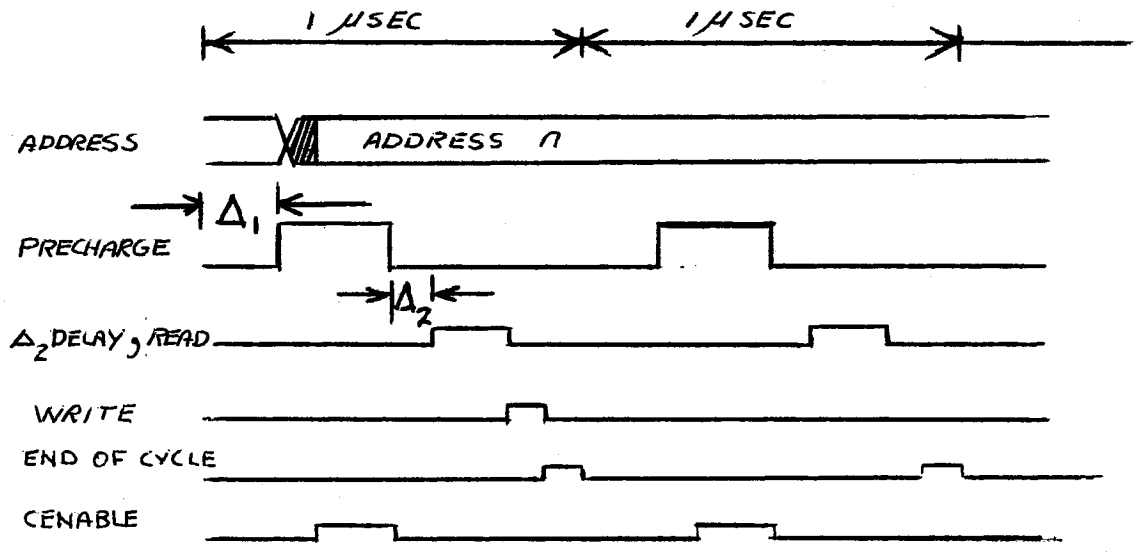


Figure 20. Timing Diagram of the Memory Level Signals

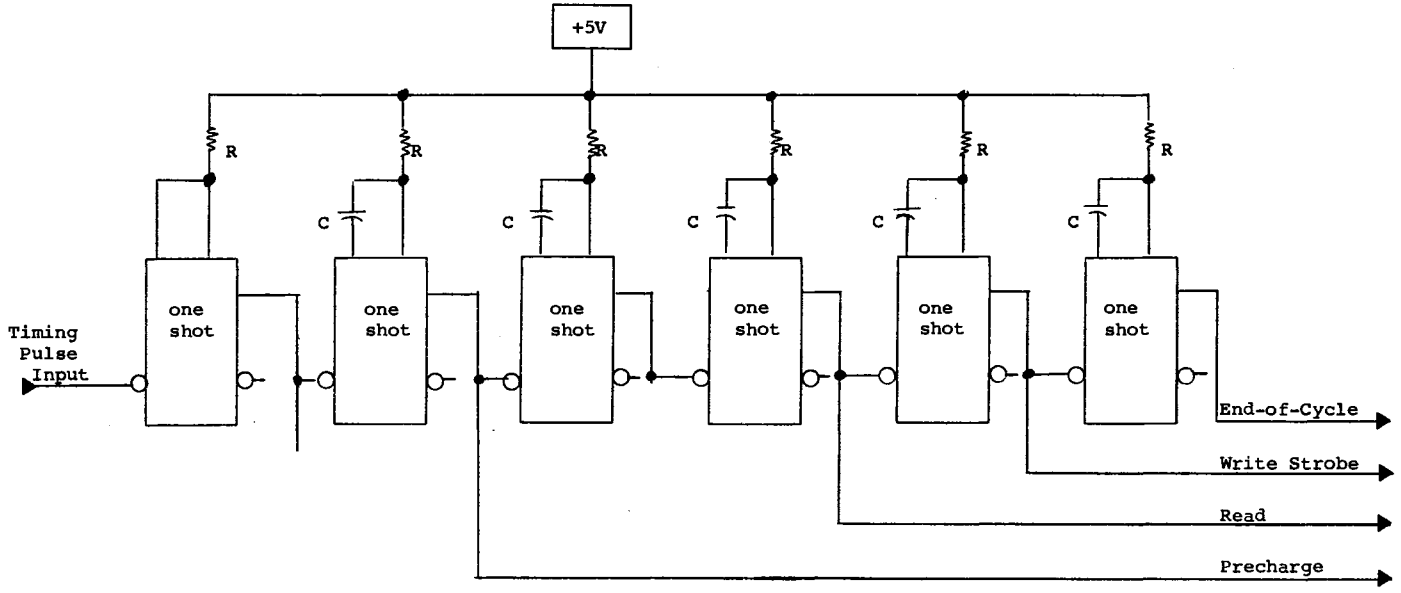


Figure 21. Schematic of the Basic Monostable Design

monostable multivibrators each sequentially triggering the following stage. The pulse widths are assigned to each monostable that satisfies both the processor's memory access and cycle requirements, yet is compatible with the Read/Write MF 7112 memory element. A worst case design is started by first assigning nominal pulse widths and then applying the $\pm$ 25 per cent tolerance for minimum and maximum worst case timing analysis. Referring to Chapter II, the minimum timing requirements of the MF 7112 are listed in Table IX.

The pulse widths and delays listed in Table X summarizes the minimum, nominal and maximum pulse widths to be expected from the monostables when the $\pm$ 25 per cent tolerances are applied. Table X was derived by first using the minimum timing requirements listed in Table IX, and then increasing this value by 25 per cent. The maximum value was derived from the nominal value in the same manner.

A memory cycle is expressed by

$$\text{Memory Cycle} = PW_1 + \Delta_2 + PW_2 + PW_3 + PW_6 \quad . \qquad (16)$$

Using the values of Table X

Minimum Cycle Time = 730 nanoseconds

Nominal Cycle Time = 975 nanoseconds

Maximum Cycle Time = 1220 nanoseconds .

A memory cycle for the system is defined in Chapter II as 2.2 microseconds. Therefore, the limitations of the maximum cycle is within the system requirements.

Mentioned at the beginning of this chapter was a caution statement referring to the refresh cycle. To qualify this, one needs to refer to

TABLE IX

MINIMUM MEMORY PULSE WIDTHS OF
THE MONOSTABLE MULTIVIBRATOR

SIGNAL NAME	PULSE NAME	PULSE WIDTH in ns
Precharge	t_1 (PW_1)	300
Δ_2	delay 2	100
Cenable	t_2 (PW_2)	200
Read	t_3 (PW_3)	180
Write	t_4 (PW_4)	100
Δ_3	delay 3	100
End of Cycle	t_5 (PW_5)	125

TABLE X

MONOSTABLE PULSE VARIATIONS

SIGNAL NAME	PULSE WIDTH	MINIMUM PULSE WIDTH IN ns	NOMINAL PULSE WIDTH IN ns	MAXIMUM PULSE WIDTH IN ns
Precharge	PW_1	300	400	500
Delay Between Precharge and Read	Δ_2	100	134	168
Cenable Delay	Δ_3	100	134	168
Cenable Width	PW_2	210	280	350
Read	PW_3	180	240	300
Write	PW_4	100	134	168
End of Cycle	PW_5	50	67	84

the latter part of this chapter; however, notice that in the monostable timing diagram of Figure 22, the End of Cycle pulse resets the refresh line which in turn changes the address multiplexer from a refresh address back to the system address. The problem area to avoid is when a cycle time is at a maximum and the End of Cycle leading edge occurs after the address register changes. With an End of Cycle pulse occurring after the MF 7112 memory elements address for the next cycle, the change will occur during the Precharge pulse. One must insure that this does not happen for data can then be transferred internally from cell to cell within the MF 7112.

Figure 23 is the final schematic of the monostable multivibrator one-shot design.

The Memory Refresh Logic

The Need for Refresh

Many methods of resupplying energy to a MOS dynamic Read/Write memory have been suggested. In all cases the purpose of the resupplied energy is to keep a capacitive node charged to the logic state intended when the memory was last altered. The name given to this resupply of energy is "The Refresh Cycle". Chapter I briefly discussed one method of internally refreshing the storage node for a two phase device. Other methods are used for single and three phase devices. In any event, the internal mechanism that implements refresh is not usually of great concern to the system memory designer. The concern lies in the method external to the memory IC. Some of the questions that arise are: Is refresh executed synchronously, or asynchronously? How often must the memory element be refreshed? What constitutes a refresh cycle? The

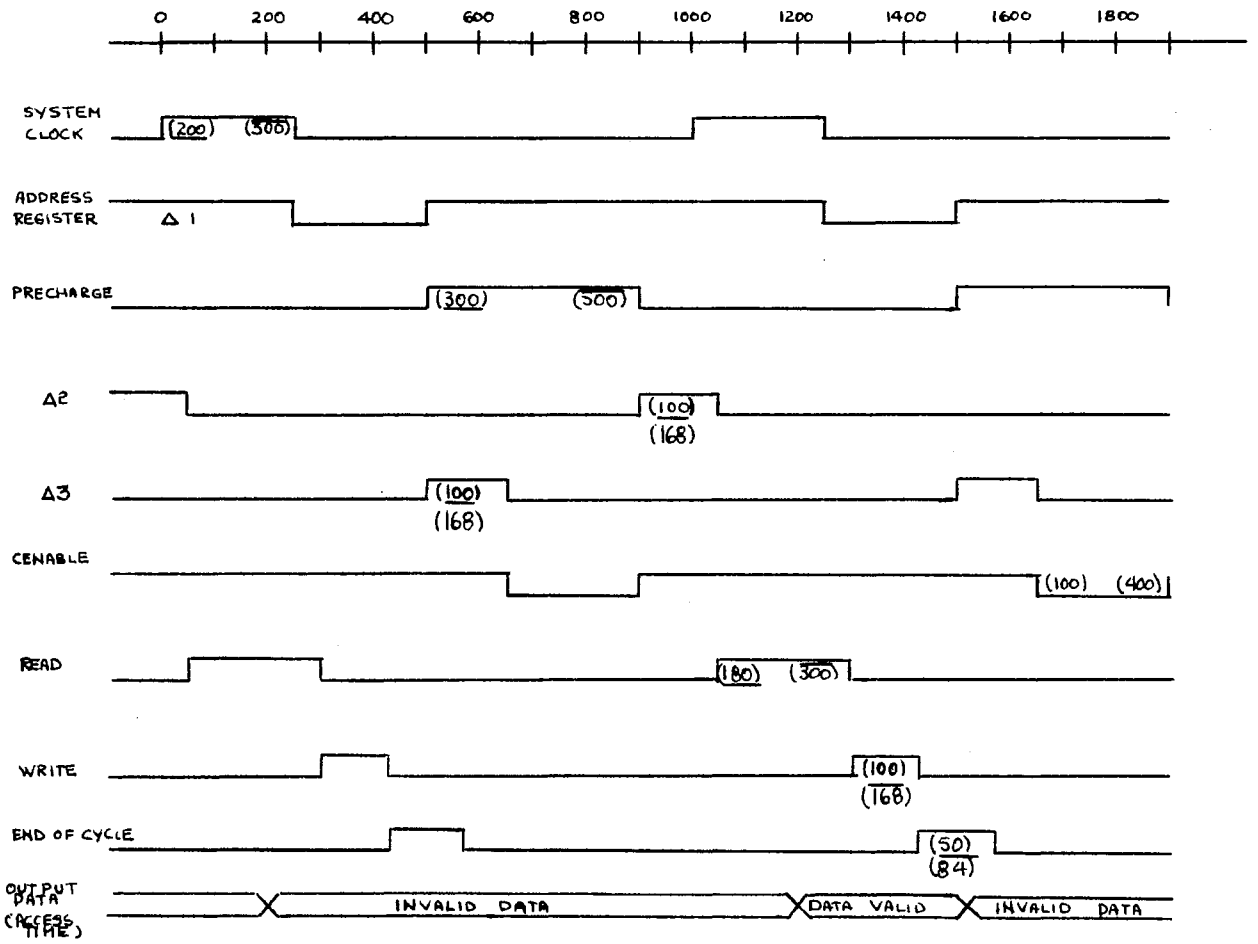


Figure 22. Timing Diagram of the Monostable Memory Cycle

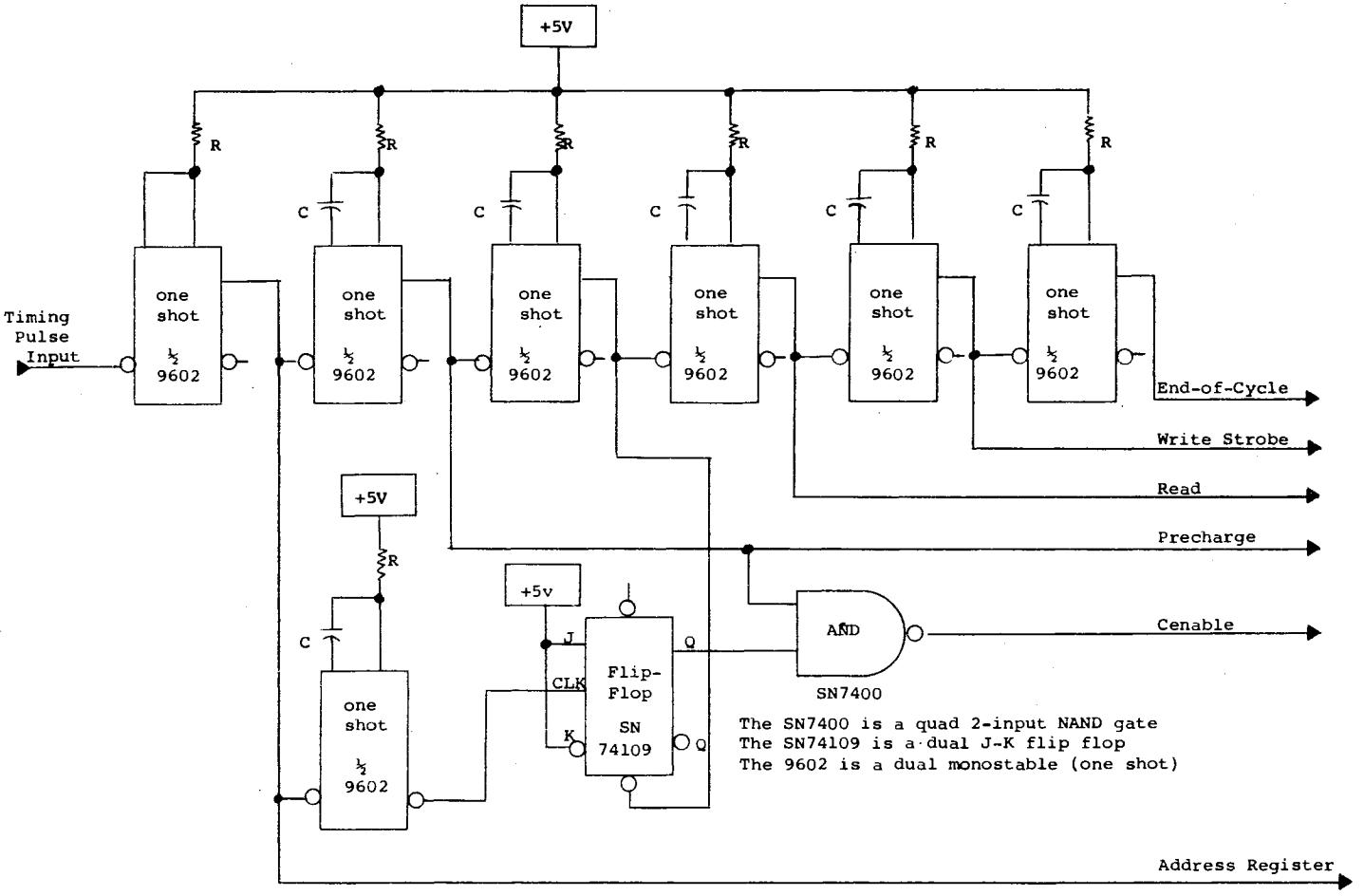


Figure 23. Final Schematic of the Monostable Multivibrator Design

answers to these questions and others are found in the type of processor and the choice of memory element. In the case of the example LSI processor and the example Read/Write random access memory IC, the MF 7112, the refresh mechanism was designed into the processor anticipating a need to use a dynamic memory element. Such may not be the case for other processor applications. Therefore, other means of mechanizing refresh are available.

The design engineer wants to weigh carefully the choice between advantages of a dynamic and a static Read/Write memory element. One of the biggest advantages of the static memory is its independence of a refresh signal. Because of the basic flip-flop design the static memory does not require refreshing.

The remaining portion of this section is devoted to guiding the engineer through a worst case design of a synchronous refresh cycle using the example processor and the MF 7112 memory IC. However, because this design is indicative of only one type of refresh method other types of refreshing will be explained also, but will not be designed into the example problem.

Types of Refresh

There are four methods of refreshing the memory element that are in use today:

1) a charge pump refresh;
2) a continuous access refresh;
3) a planar refresh; and
4) a processor controlled refresh.

The example design to follow will be the Processor Controlled

Refresh. The Processor Controlled Refresh was chosen because it is the more difficult of the four methods to implement. Each of the three other methods is an adaptation of the Processor Controlled Refresh.

Before continuing with the design aspects, the four refresh methods will be examined on a device level.

The theory of the Charge Pump was developed in the General Electric research laboratories in the late 1960's. However, the technique was never applied to production memories until the development by American Memory System (AMS) in 1972. The first memory product is now entering the market place in the fall of 1973 and is designated the AMS 7001.

The Charge Pump in the true sense does not require a refresh signal. An asynchronous signal in the form of a sinusoidal, triangular, or square wave is applied to the charge pump input of each memory IC. This is the refresh mechanism and no other method of refreshing is required. Accessing either in a Read or a Write cycle is done completely independent of the charge pump signal. Because of this, the charge pump method is the easiest of the refreshing techniques to use. However, the disadvantages of the charge pump refresh method lies in the physical size required to make a storage cell. Thus far, the processing techniques have restricted memory size to 1024 bits or less. Figure 24 is a schematic of the basic charge pump and storage cell for one bit. Memory sizes of 4096 or larger becomes prohibitive because of the large silicon area required.

It is doubtful that the charge pump method will be applied to memories larger than 2048 bits because of the cell size limitations. However, if a technological breakthrough does happen the charge pump method is more attractive than other refreshing methods.

Each of the three remaining refresh methods occur during a pseudo

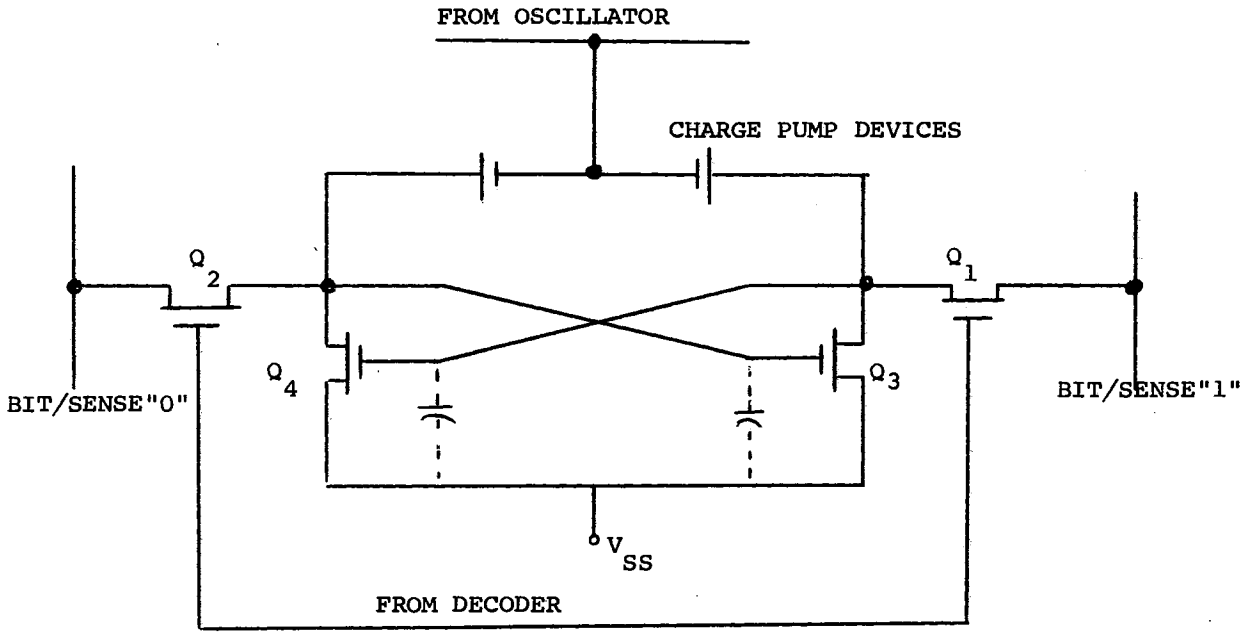


Figure 24. Schematic Representation of the Basic Charge Pump Storage Cell

Read cycle. A refresh of the stored energy to the storage node occurs during the "off" time between the clocks. By causing a Read or a Fetch command to occur, one portion of the memory has been refreshed. For example, referring to Figures 25 and 26, each of these types of memory elements are configured into a X-decode and a Y-decode with each decode connected to the storage matrix. Refresh is accomplished on a row basis. Therefore, when a Read Cycle is completed for any one address a refresh cycle has been accomplished for the entire row of storage cells selected by the address. To insure that the entire memory has been refreshed the least significant bits (LSB) of the address must contain all 32 possible combinations for the 1024 bit memory element of Figure 25. In a like manner, the four LSB of the address must contain all possible 16 combinations for the MF 7112. All combinations must be accomplished within the refresh cycle time specified by the memory IC manufacturer. For both the 1103 memory IC and the MF 7112 memory IC, this time for all possible combinations is 2 milliseconds. If the design can guarantee that all combinations of the X-decode LSB's are exercised within 2 milliseconds then this is called Continuous Address Refresh. Only in very specialized application is it possible to guarantee that all X-decode combinations are used for a random access memory.

For this reason the two remaining methods of refreshing are generally used in a random access memory. The Planar Refresh is a method whereby all lines of the X-decode to the internal matrix are turned on and an internal refresh occurs. The Planar Refresh is a synchronous method and must be accomplished during the processor non-memory access time. Figure 27 shows the internal memory connection for a Planar Refresh IC.

The internal Planar Refresh requires extra silicon area on the

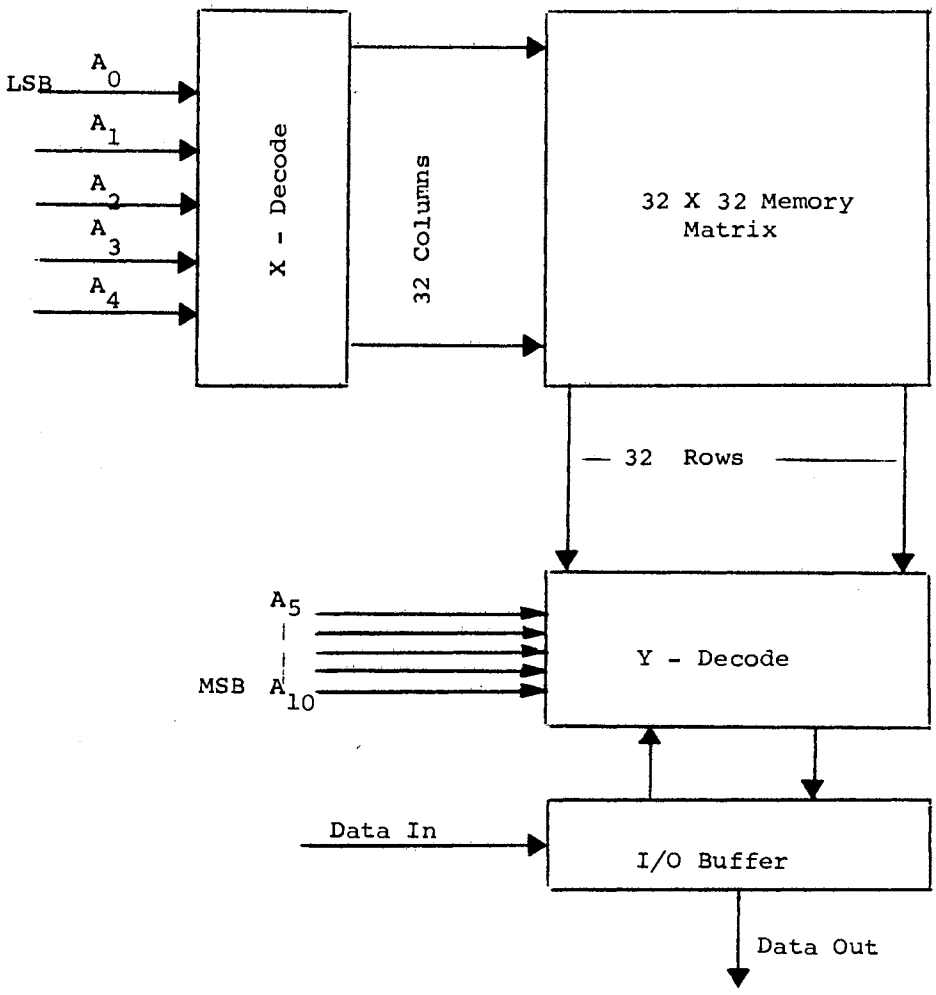


Figure 25. A Block Diagram of a 1024 Bit Semiconductor Memory IC Using Refresh by Addressing

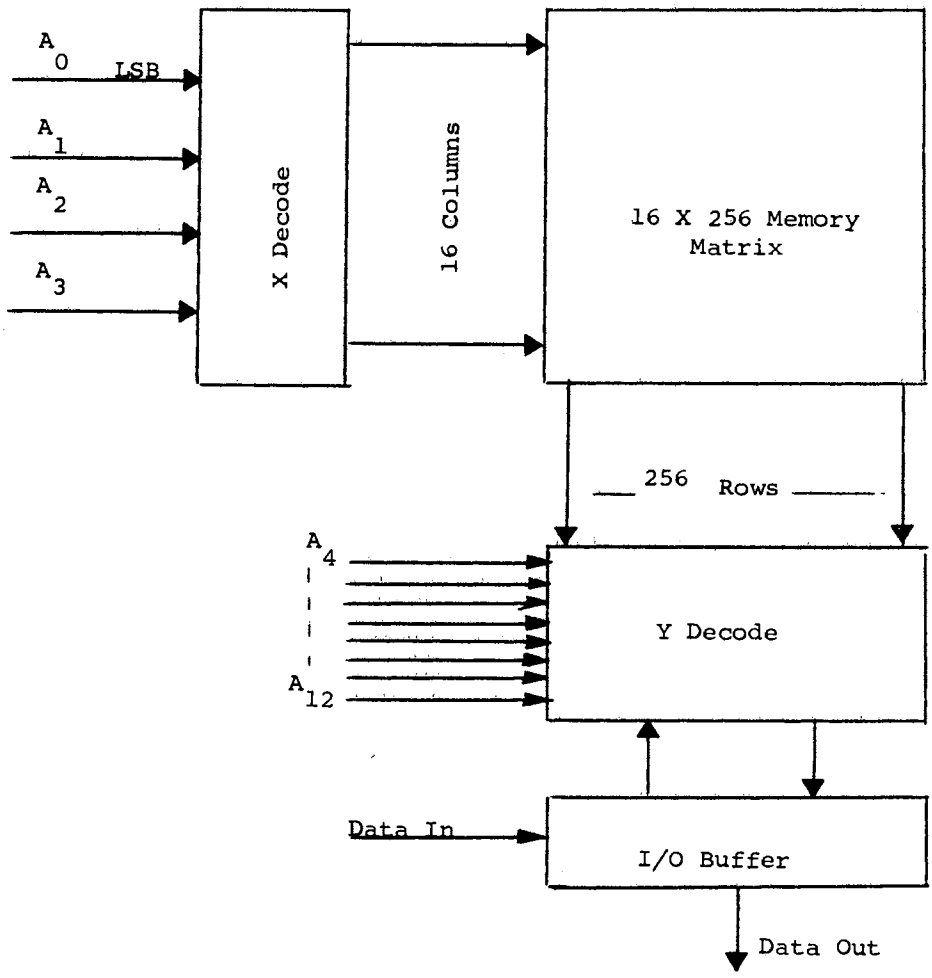


Figure 26. A Block Diagram of a 4096 Bit Semiconductor Memory IC Using Refresh by Addressing

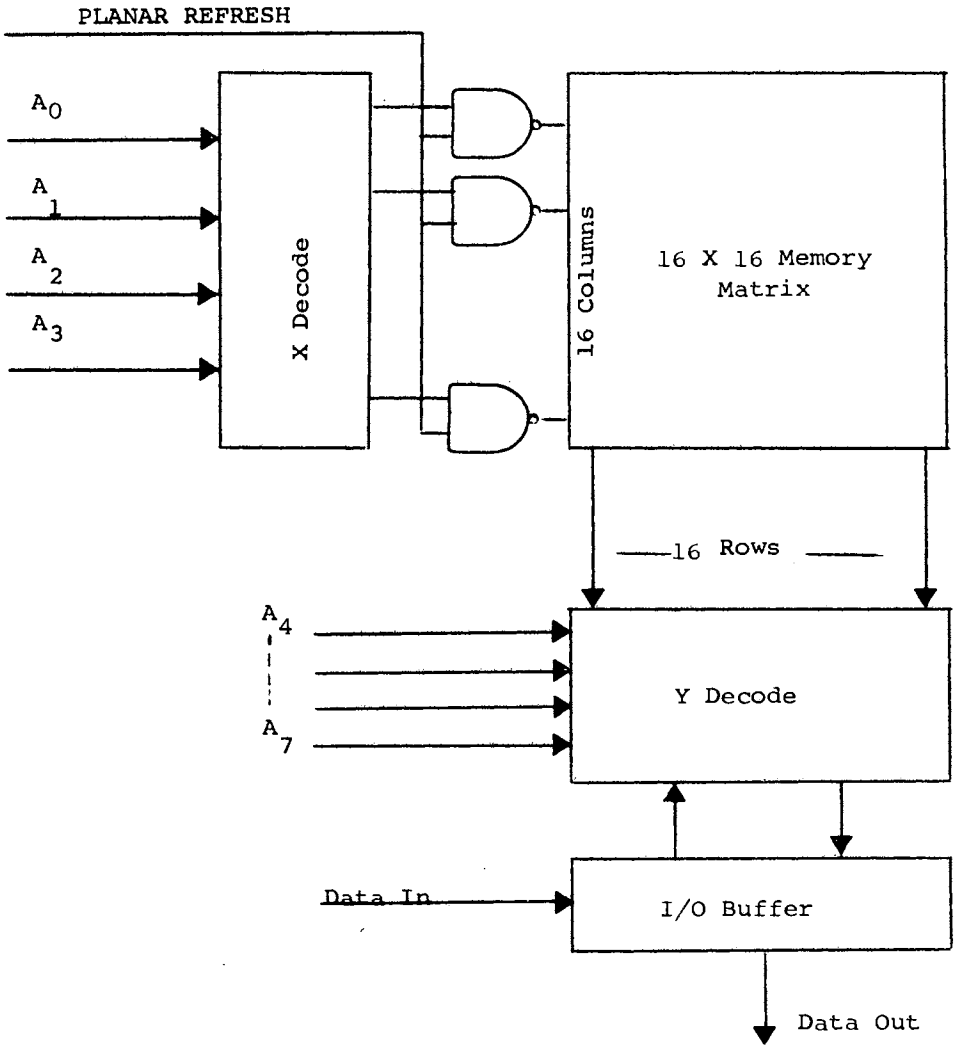


Figure 27. A Block Diagram of Planar Refresh

memory die and is often left off the IC because of area restrictions. However, this method is becoming quite popular and will probably replace the refreshing through use of the LSB address lines. Notice that for applications that can guarantee the Continuous Access Refresh method, the Planar signal can be inhibited and the memory refreshed through the LSB, X-decode address lines.

The final method of refresh utilizes a processor controlled refresh signal that causes a counter to update a refresh address each time the control unit signifies that a refresh cycle is to take place. The period of refresh is generally established based on the processor's instruction time. Duration of the refresh is generally established as one period of the processor central clock. This method is expanded in detail in the following paragraphs.

Operating Conditions

Processor Operation

There are two operating conditions that the designer must consider. The first of these is the condition the processor can assume during normal operation. The second is the memory operating condition. These conditions are reviewed below. Each of these conditions are important because they have a direct relation to the refresh interface circuitry. Any processor can have any, or all of these conditions. For the purpose of this design example, it is assumed that the LSI processor has all possible operating conditions. The four possible operating conditions are:

Condition IA, Normal Operation. This is the normal run condition of the processor. The central clock is running uninhibited. Arithmetic and control unit indicate a non-halt condition.

Condition IB, Normal Operation, Refresh Requested. This is the same condition as IA except the control unit has determined that a refresh condition exists during a particular instruction time.

Condition II, Repair Mode. Condition II is a processor test mode primarily used for software debug or hardware repair. This condition is a processor single instruction mode whereby the operator can step through the instructions and execute one instruction at a time.

The memory design must account for this mode. An external signal (Halt) originating from the control unit becomes active and the Refresh signal becomes inactive by virtue of the processor being halted. These are the operations that occur:

1) the central clock is running as if the processor is in the normal operating mode (Condition I);

2) the Refresh signal becomes inactive, a low logical state;

3) a Halt signal is activated by the control unit. The purpose of this signal line is to indicate that the processor has halted operation. During the single instruction, the Halt signal indicates the processor has returned to Condition I for the duration of the instruction.

Condition III, Microprogramming Level. Condition III allows the operator to select an even lower level of processor control by allowing the central clock to be stopped and the processor to be controlled by a single clock mode on a micro-instruction level.

A single clock mode is difficult for a semiconductor memory to

maintain the stored data because none of the essential interface signals from the control unit are now operative. With the central clock stopped neither of the two timing methods described in the beginning of this chapter will operate. Therefore, a special set of logic must be included to meet the memories timing and refresh needs and still operate in a single clock mode. Activation of the clock is assumed to be arbitrary depending on the desires of the operator. The special method of working with the single clock for this design example will be to gate the clock and detect a clock activation when the processor is in Condition III. Figure 28 represents one method that can be used to gate the clock for single clock operation. Also included in Figure 28 is a brief timing diagram of the operation.

The operating conditions that can now be assumed are:

1) the ungated clock operating normally;

2) the gated clock is a single pulse as shown in Figure 28;

3) the Halt signal in relation to the gated clock is a pulse as shown in Figure 28; and

4) the processor Refresh signal will remain inactive.

Condition IV, Power-Off State, Memory to Retain Data. Condition IV is the power-off/battery operation mode. The condition required for the logic is to constantly refresh the memory within the maximum allowable time as allowed by the memory manufacturer. All processor signal leads must be in a high impedance state. The central master clock is not available in this mode. Power is supplied from batteries. Refresh is initiated from a source powered separately from the processor.

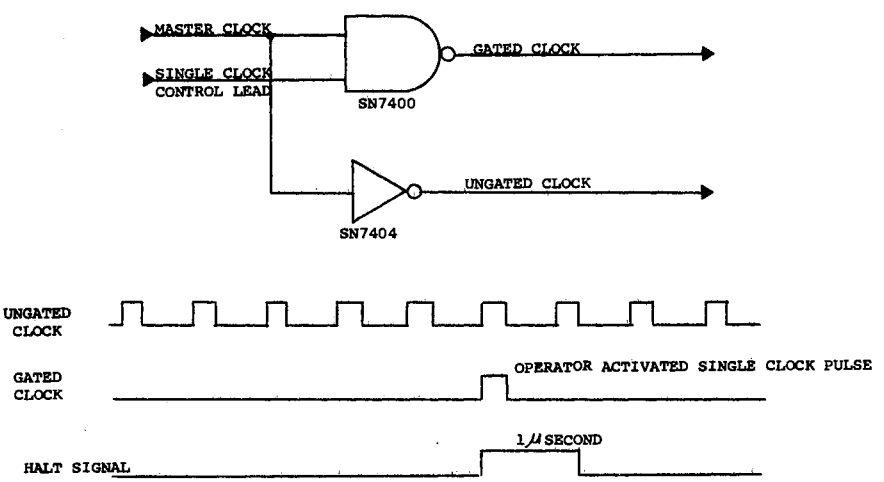


Figure 28. A Method to Gate or not to Gate the Master System Clock

Memory Element Operation

The previous discussion centered about the processor operating conditions. This section describes what conditions a memory must assume regardless of the processor's condition.

A dynamic IC Read/Write memory will require refreshing. The MF 7112 is a dynamic memory element requiring regreshing with 2 milliseconds for the entire memory matrix. The MF 7112 can be refreshed using either the processor controlled or the continuous address refresh method. The organization of the MF 7112 was reviewed previously and shown in Figure 26. The memory requires the least four significant bits of the address register to completely refresh the memory matrix. One of the simplest methods to address all possibilities is to sequentially progress through each binary possibility. Table XI represents the necessary address at refresh time t.

As discussed in this chapter under Types of Refresh, each refresh cycle is a pseudo Read cycle, but the address is pointed to one of the refresh addresses shown in Table XI.

Regardless of processor operating conditions, or other factors, the memory must be refreshed in accordance with the inequality

$$t_0 + t_1 + t_2 + \cdots + t_{14} + t_{15} < 2.0 \text{ milliseconds} \quad . \qquad (17)$$

The refresh cycles are then repeated in accordance with the inequality or stored memory data will be in error.

TABLE XI

LOGIC TRUTH TABLE OF THE REFRESH ADDRESS REQUIREMENTS

LSB AR16	AR15	AR14	AR13	REFRESH TIME
0	0	0	0	t_0
1	0	0	0	t_1
0	1	0	0	t_2
1	1	0	0	t_3
0	0	1	0	t_4
1	0	1	0	t_5
0	1	1	0	t_6
1	1	1	0	t_7
0	0	0	1	t_8
1	0	0	1	t_9
0	1	0	1	t_{10}
1	1	0	1	t_{11}
0	0	1	1	t_{12}
1	0	1	1	t_{13}
0	1	1	1	t_{14}
1	1	1	1	t_{15}

Implementing the Processor Controlled Refresh

Requirements Review

The requirements of the processor are: (1) provide a synchronous refresh signal to the memory interface logic for each instruction period; (2) provide the memory interface logic a signal indicating a Halt condition; (3) supply a gated and ungated clock to distinguish between a single instruction and a single clock condition. In addition to the three preceding requirements, which can be called the Processor Refresh Requirements (PRR) the control unit must provide the normal address, data, and Write control signals. Figure 29 represents a block diagram of the PRR.

The interface logic insures that refresh occurs properly without interfering with the normal function of the memory. The interface logic must also operate under any of the four processor operating conditions.

Realizing the Refresh Addressing

The first task is to design the logic that transmits the address register contents to the memory, but replaces the refresh address for the instruction address at Refresh time as indicated by the control unit. Two logical functions must be accounted for: (1) a means of substituting the control units address register contents with a refresh address and, (2) a means of generating the refresh address and updating this address on command from the control unit.

Remembering that the refresh address operates on the 4 LSB of address (the X-decode inputs of the MF 7112) a modulo 16, 4 bit, up-counter would accomplish the desired refresh address generation. To

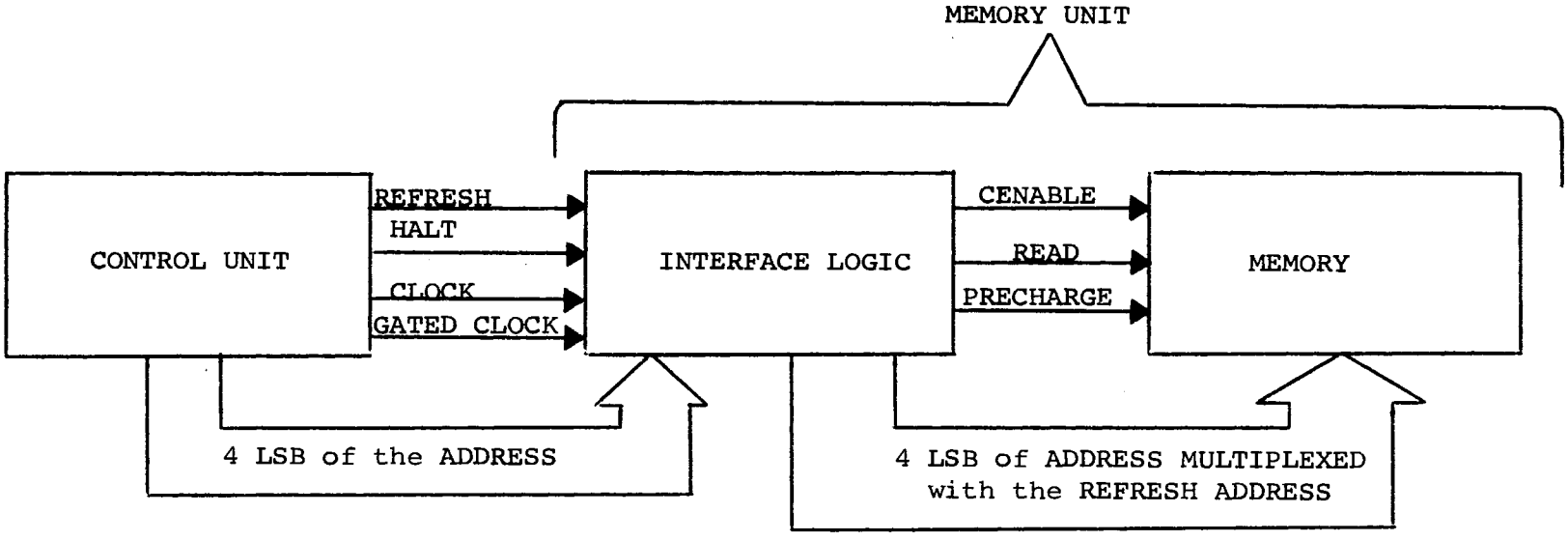


Figure 29. A Block Diagram of the Processor Refresh Requirements

substitute the control unit's address registers contents for a one (1) microsecond Read cycle with a refresh address a quad two-input multiplexer would accomplish the desired logic and will operate on the control unit's command. Later it will be shown that the change occurs on command from the interface logic which in turn operates from the control unit. Figure 30 represents the complete schematic of the refresh address scheme. A SN 74193, modulo 16 up-counter and a SN 74157 multiplexer are used to implement the function.

Implementing for Conditions II and III are the most difficult to account for because of the asynchronous timing of the operator arbitrarily selecting an instruction command. For both Conditions II and III the gated clock will be forced to a D.C. quiesent state during the period of no activity. The gated clock will be forced into a normal active mode by the control unit. Therefore, the gated clock can be used as a signal line that signifies the normal inactive state of refresh for Conditions II and III has become an active state which could be either a memory alter (Write) or a memory fetch (READ). At first glance the Halt signal seems to meet the requirements to detect an active state for Conditions II and III; however, it is often true that the Halt signal is derived from the decode logic of the control unit and will become inactive, i.e., go to a logical 0 state, for an instruction time. This means it will not change states for all conditions of Condition III. Because of this, another type detector will be required, sensing that a memory active state is required using the gated clock as the signal line. One method of determining the state of the clock is to force the clock pulse, in this case 250 nanoseconds nominal, to charge a capacitor over a time constant equal to about 5 times the clock period. If the ungated clock

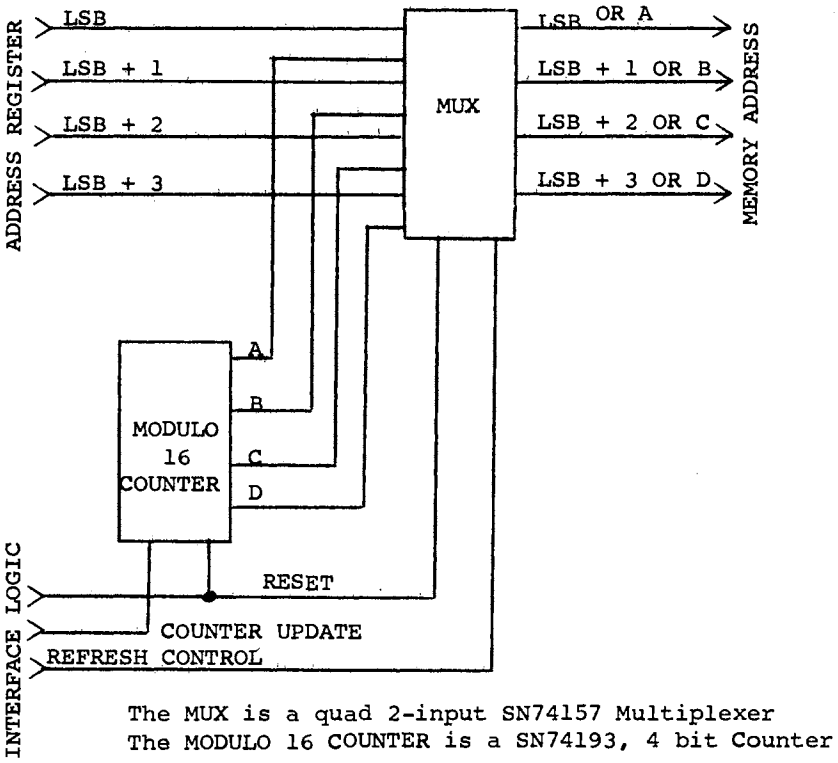


Figure 30. A Completed Schematic of the Refresh Address Scheme

stops running, as in Conditions II and III, the clock detector will assume a logical state of "1". When the clock is running, as in Condition I, the detector assumes the complement state. This is also an inactive state in that the detector is in a static logical "1" or "0" as long as the clock is running or if the clock has been stopped. This detector will be discussed in detail in the next section.

Before continuing with the design a review of the requirements and logical states of the control signals from the control unit is in order. Special needs of any processor can be added as required. These requirements and logic states are:

1) when the processor is in a Halt (Conditions II and III) the refresh multiplex lead is to indicate a Refresh state;

2) a halt mode must be detected using both the Halt signal lead and the ungated clock detector;

3) the refresh signal from the control unit is to be a one (1) microsecond positive pulse, synchronous with the master clock when the processor is in Condition I;

4) battery operation signal to be a logical "1" (high) when inactive and must originate outside of the control or arithmetic units;

5) the ungated clock to run continuously except for Condition IV;

6) all logic signals not involved with operation of the memory during Condition IV are to assume a high impedance; this will prevent them from interfering with the logic needed to operate the refresh logic;

7) for Condition I, during an instruction time, the control unit cannot require other operations concurrently and simultaneously with the refresh command, e.g., a memory Read or Write command; and

8) sixteen complete memory refresh cycles must be completed within

2 milliseconds regardless of the condition of the arithmetic or control units. As the refresh pulse from the control unit originates every instruction period regardless of the instruction length, the software must be arranged such that a minimum of 16 refresh commands are initiated within a 2 millisecond period. Failure to do this could lead to loss of memory contents.

A generalized processor instruction/refresh timing relationship is shown in Figure 31.

With the basic system's refresh processes and processor operations defined, one must now consider the worst case conditions of Refresh pulse with respect to the master clock. Figure 32 represents this relationship for the LSI processor of the example problem.

When confronted with a minimum and maximum spread between the two extremes as shown in Figure 32, it is always better to resynchronize the pulse with the remainder of the timing interface. The large extremes between the minimum and maximum "true" times of the processor controlled refresh signal is caused by internal delays within the control unit. These delays are caused by processing, environmental, and power supply effects. The timing of Figure 32 does not mean that the same control unit will exhibit these types of delays, but rather, unit A could have a minimum delay time while unit B could have a maximum delay time under the same conditions. Therefore, the precaution in a worst case design is to insure that all delay times are accounted for.

The interface control signal to be developed, called RC for reference purposes, can be expressed as

$$j = (R \cdot y_1)t_1 \tag{18}$$

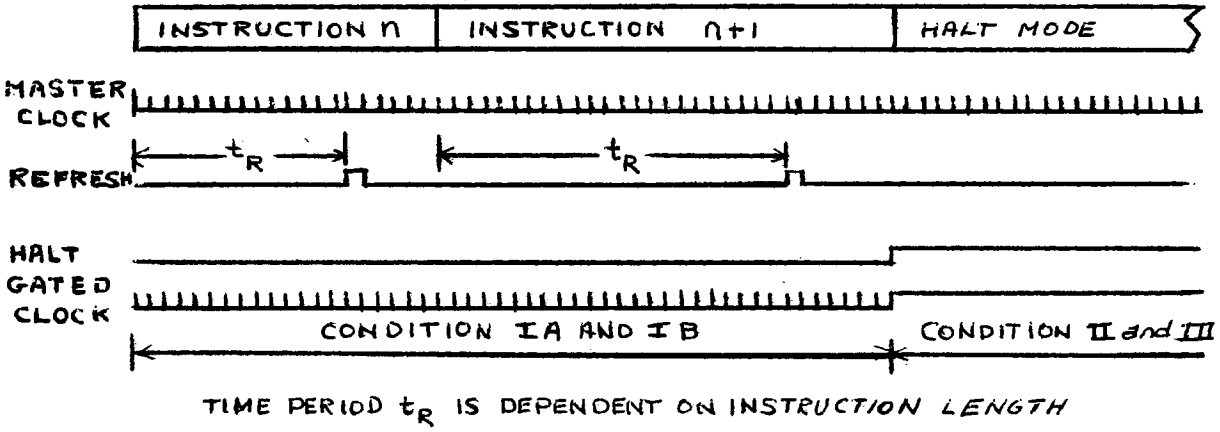


Figure 31. A Timing Diagram of the Processor Controlled Refresh

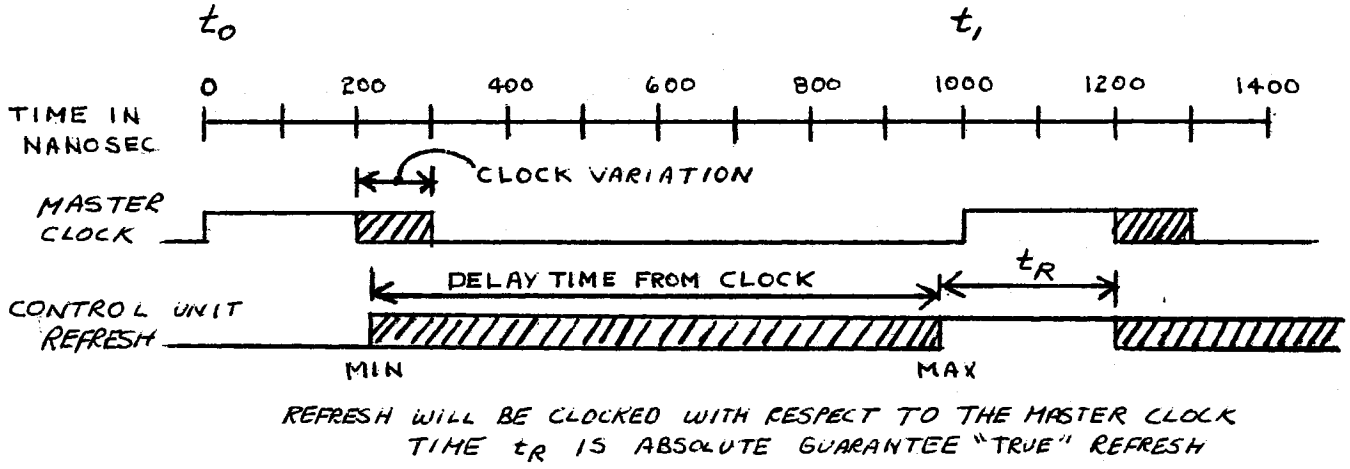


Figure 32. A Worst Case Timing Diagram of the Refresh Pulse

$$k = (y_1)t_1 \tag{19}$$

which is the input expression for a triggered j-k flip flop. Time t_1 is the leading edge of the master clock as shown in Figure 32. One must check to insure that the movement in the time frames of the Refresh pulse does not interfere with the performance or operation of the memory. For instance, on short duration instructions, the next instruction address change could occur immediately after the Refresh pulse. Implementation of the flip flop input equations are shown in Figure 33.

Implementing the Clock Detector

If the processor was never operated in the Single Instruction or Single Clock Conditions, then the refresh logic design would be complete. However, in order to account for Conditions II and III more logic is required. The Clock Detector was first discussed on page 69. The logical OR of the gated clock, detected by the Clock Detector, and the Halt signal (H) will satisfy both processor conditions. Figure 34 is the schematic for this connection. The output function of the gated clock whose state is detected, and the OR of H are expressed by

$$f = \overline{CD} + H \quad . \tag{20}$$

This function will be used to expand the refresh expression later in this discussion.

In order to provide the clocking source for processor Conditions III and IV, an astable will be required. The period of the astable is selected to be 125 microseconds. With this period the total of 16 refresh address states initiated by the astable keeps the memory refresh period less than 2 milliseconds. Possible astables suitable for use in

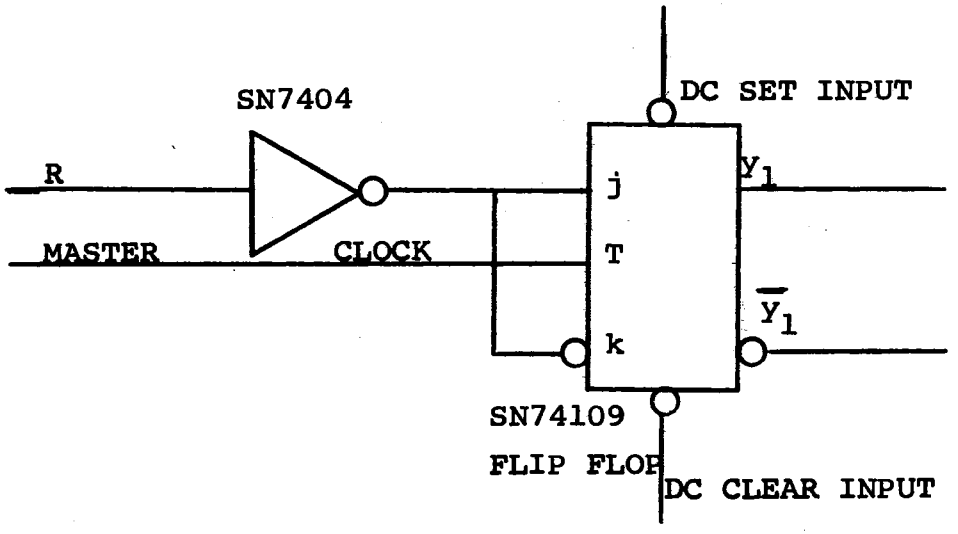


Figure 33. Implementation of the Logic to Synchronized Refresh Signals

this application can be found in the appendix.

The astable will be asynchronous if allowed to free run. Two alternatives present themselves when implementing the astable. The designer has the choice of synchronizing the astable to the logic or using the logic to synchronize the output of the astable. The latter is chosen generally because of the degree of difficulty in worst case designing a synchronous astable. Synchronization is required only for Conditions II and III. Asynchronization is permissible for Condition IV as the arithmetic and control units are in a power-down mode. The logic signal developed in Figure 34 will be used to control the passage of the astables output pulse to the j-k flip flop shown in Figure 33. In order to synchronize the astable to the ungated clock (the flip flop in Figure 33 is triggered by this clock) an S-R flip flop will be used. The input equation for the flip flop is expressed as

$$s = R \cdot \overline{f} \cdot \overline{y}_2 \quad . \tag{21}$$

The equation can also be expressed as

$$s = R \cdot CD \cdot H \cdot \overline{y}_2 \quad . \tag{22}$$

In order to synchronize the astables output pulse the astable can be logically ORed with the set input of the S-R flip flop. This is expressed as

$$s = R \cdot CD \cdot \overline{H} \cdot \overline{y}_2 + \text{astable} \quad . \tag{23}$$

Rewritten, this equation can be expressed as

$$s = R \cdot CD \cdot \overline{H} \cdot \overline{y}_2 + A \tag{24}$$

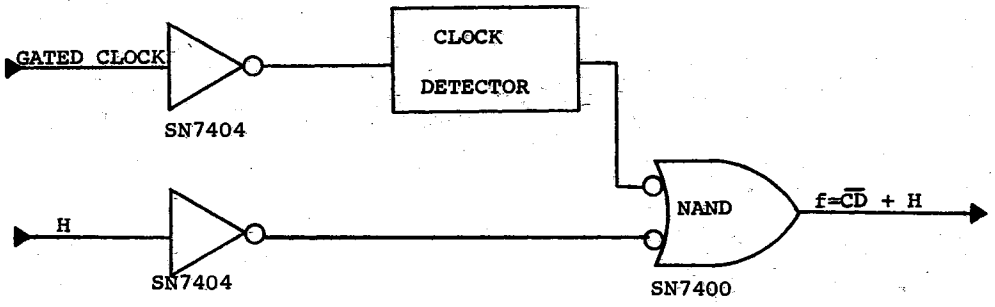


Figure 34. A Logic Diagram Representing a Detected Gated Clock

To insure that the astable inhibited for single instruction or single clock (Conditions II or III) the astable pulse is inhibited by a logical AND of the Halt (H) and the Clock Detectors output. This can be expressed and expanded as

$$s = R \cdot CD \cdot \overline{H} \cdot \overline{y}_2 + (A \cdot \overline{CD} \cdot H \cdot \overline{y}_2) \quad . \qquad (25)$$

To reset the S-R flip flop the input r must be pulsed with a signal that will occur after each cycle yet allow the refresh signal to propagate to the j-k flip flop previously used to insure the inteface logic detects the processor controlled refresh signal. The Cenable memory clock will serve this purpose.

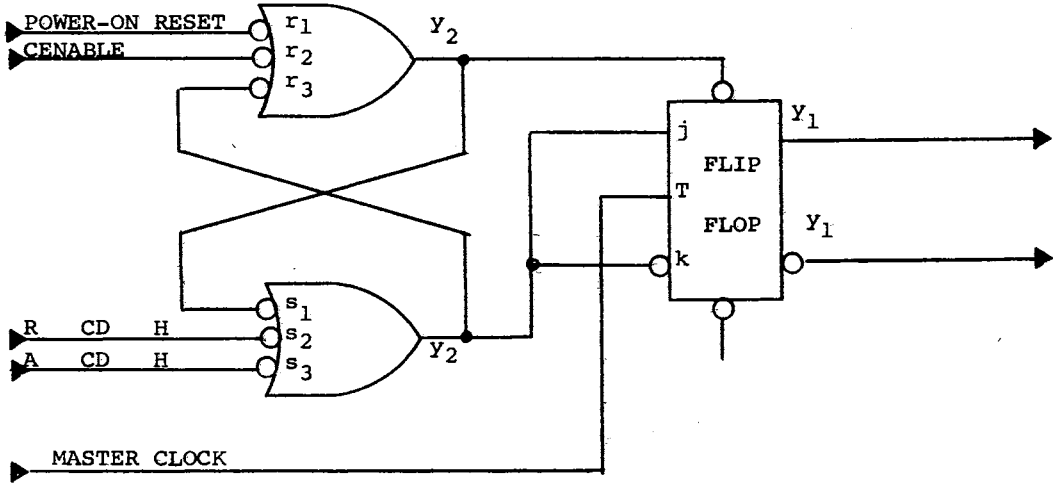
The final signal to the S-R flip flop is the initialization power turn-on signal. This insures the memory initializes in a non-refresh condition for immediate memory use.

The r input can now be written as

$$r = \text{Cenable} \cdot y_2 + \text{power-on} \cdot y_2 \quad . \qquad (26)$$

The implementation is shown in Figure 35. Also included is the j-k flip flop of Figure 33.

The final condition to account for is the power-off Condition IV. To detect this condition the circuitry must first anticipate a power down situation and signal the logic to stop memory and processor operation. This signal must also cause an unconditional refresh state to occur. This special signal is the key to a power down mode. The unconditional refresh is performed by entering the j-k flip flop by means of the D.C. static inputs. This will force the interface logic to slave the astable and ignore processor signals until the power down signal is removed.



The 3 input NOR gates are connected as a S-R Flip Flop and are made up of two modules of a SN7410.

The flip-flop is ½ of a SN74109 j-k flip-flop

Figure 35. A Schematic Representing the Refresh Interface Logic Including Astable Synchronization

The final configuration of the refresh logic is shown schematically in Figure 36.

Detecting the Halt Mode

The previous section made use of a clock detector circuit to distinguish the difference between a single instruction (Condition II) and a single clock (Condition III). The intent of this section is to describe a method of accomplishing the detector.

As discussed previously, the purpose of this detector is to signal when the clock has stopped running. A detector circuit that has operated satisfactorily is shown schematically in Figure 37.

The gated clock signal is buffered by a SN 7400 TTL inverter which in turn couples into a discrete transistor network. The discrete network is used instead of an integrated circuit to better guarantee the charge-discharge rate of the capacitor C. Transistor Q_1 switches at a rate set by the gated master one megahertz clock. However, transistor Q_2 is kept cut-off as the charge rate, τ, of capacitor C does not allow the transistor base voltage of Q_2 to reach a condition whereby Q_2 saturates. This is true as long as the clock continues to run. When the clock stops, the capacitor C charges at a rate τ which can be set 5 times the period of the clock. This will in turn cause transistor Q_2 to saturate indicating that the clock has stopped running and that processor Condition III exists.

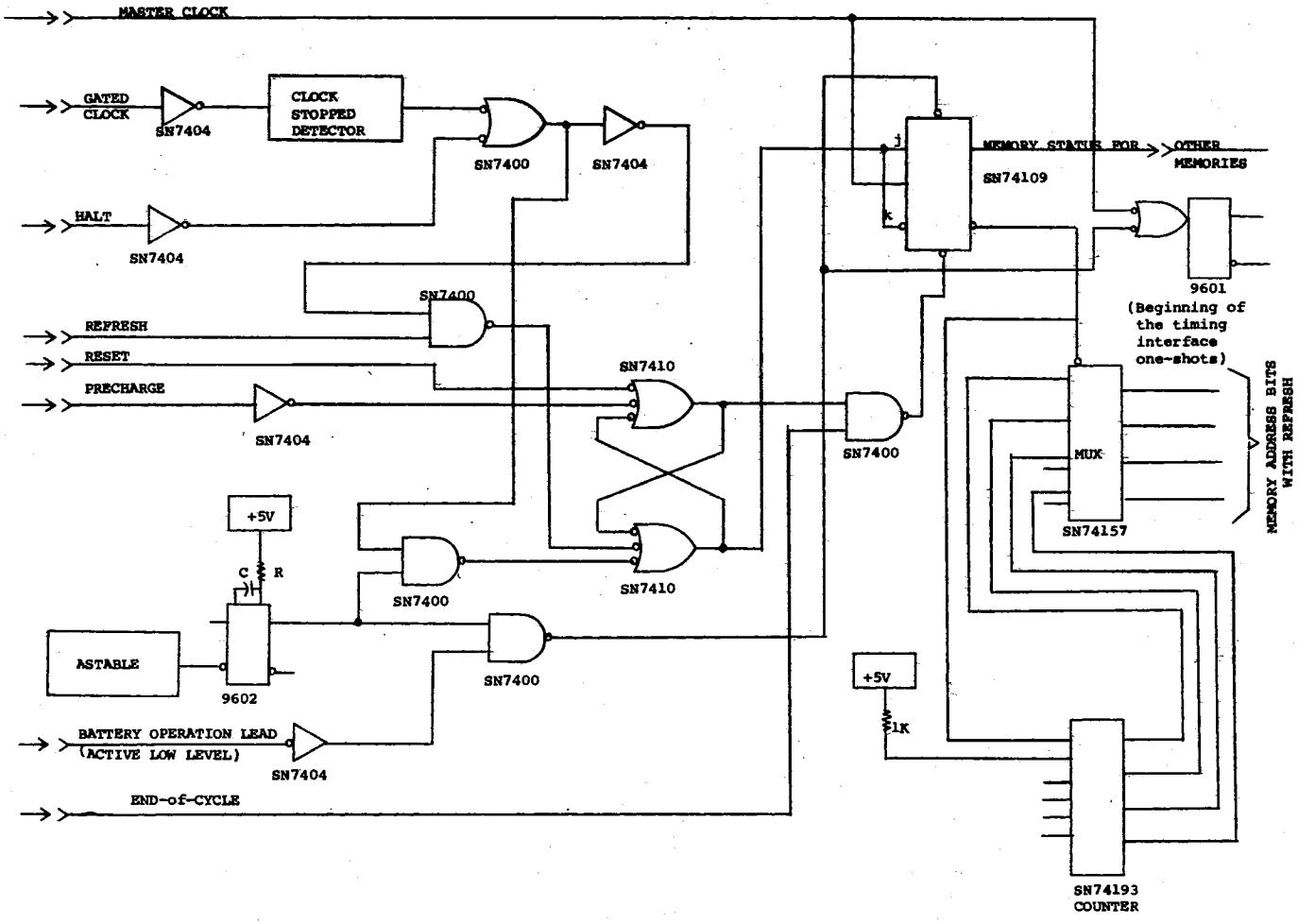


Figure 36. Schematic of the Final Overall Refresh Interface Logic

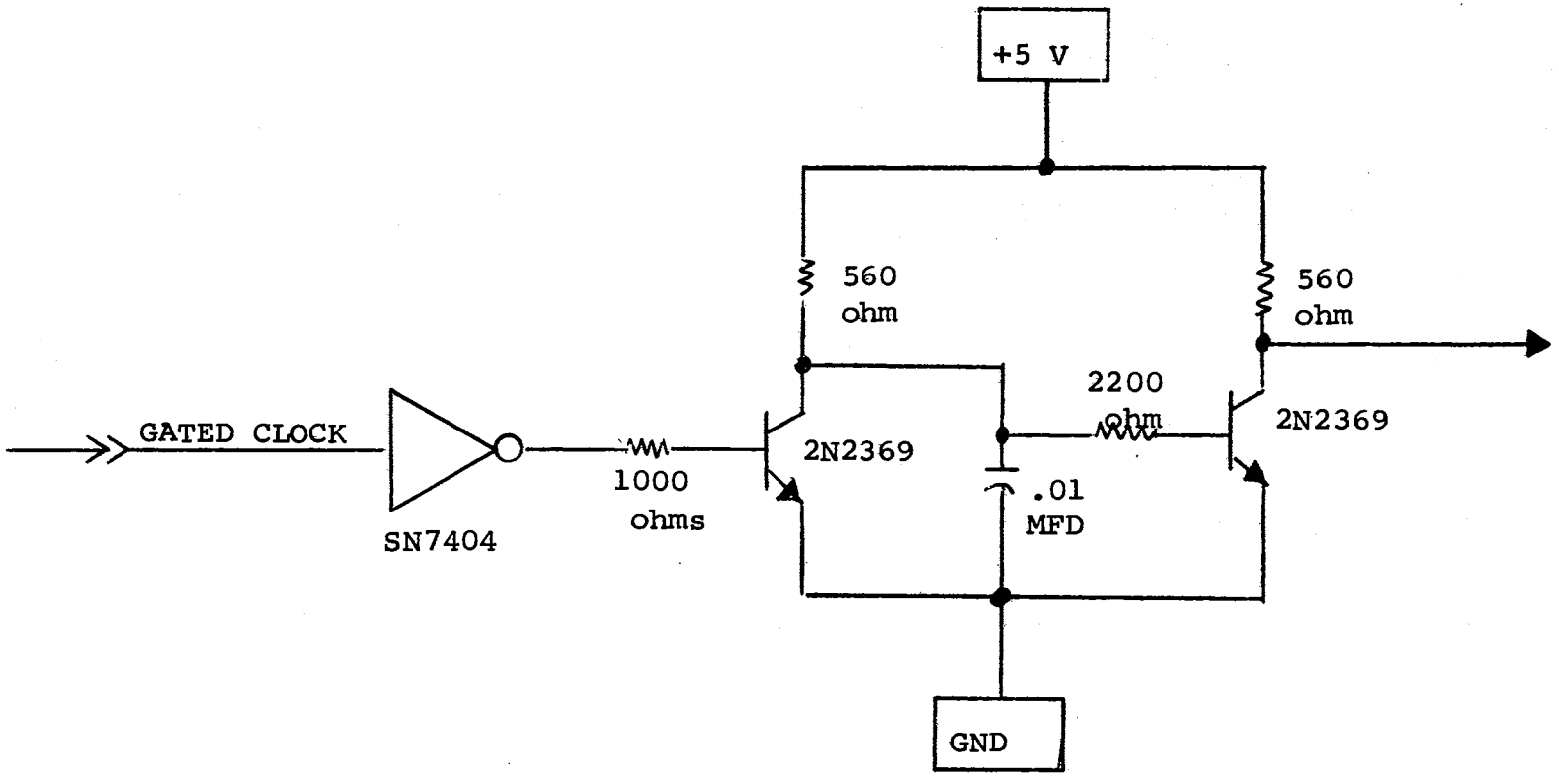


Figure 37. A Schematic of a Possible Clock Detector

The Write Interface Logic

Introduction

To alter the contents of memory the control unit will command what is commonly called a Write cycle or Alter cycle. In instruction language this is sometimes referred to as "store immediate". The alter command will consist of a pulse or series of pulses synchronized with the central master clock.

The purpose of this section is to develop the interface logic such that the processor can properly interface with the MF 7112 memory element. The interface logic is normally required, even for static memories, in order to account for internal circuit delays and the precise timing necessary to alter the contents of memory. The two main topics of this chapter have consisted of the timing logic and the refresh logic. Figure 38 shows the representation of the three interface networks, including the Write interface.

Defining the Write Cycle

The Write cycle of the system is predetermined by the instruction, decode and execution logic of the control unit. Occurrence of a store immediate command is a function of the software, while the control unit's write strobe is a function of the inherent delays within the logic of the control unit. The write strobe originating at the processor and sent to the memory should first be defined by statements and a timing diagram. The following is a set of definitions for the example processor.

1) Occurrence of the Write command is arbitrary depending on the program requirements of the software.

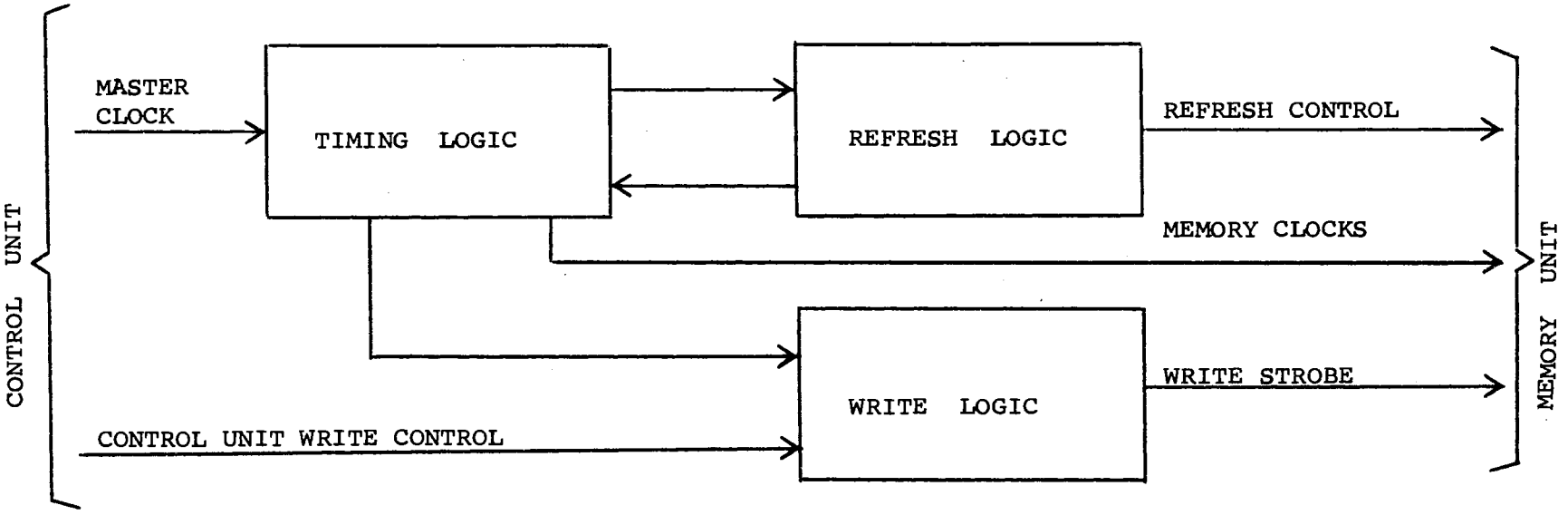


Figure 38. A Block Diagram of the Timing, Refresh, and Write Interface Logic

2) Period of the Write strobe from the processor is dependent on the instruction format which in turn itself is variable in length.

3) Duration of the write strobe from the processor is always one microsecond independent of instruction length or any of the three conditions described in the refresh section.

4) Processor Refresh and Write never occur simultaneously within any given instruction.

5) Time duration from a store immediate to the next instruction address change will be 2 microseconds minimum.

6) A read after write from the arithmetic and control unit is not required during an instruction.

7) A read before write is always required.

8) Output data from the memory is not valid during a Write cycle.

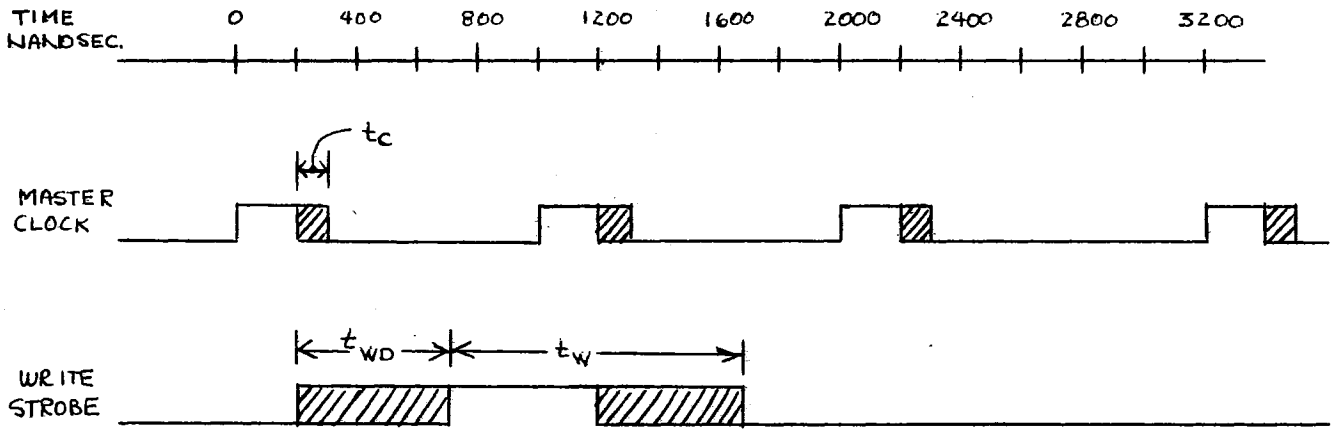
9) In the worst case analysis, if the rising edge of the Write strobe (the leading edge) is at a maximum propagation time, then the falling edge of the pulse will be at a maximum.

10) A Write strobe will not occur in the processor Condition III.

11) The Write strobe to the memory element is "true" in the logical 1 level and must be properly placed within the events of a memory write cycle.

A Description of the Timing

From the timing relationship shown in Figure 39, it will be noted that the Write strobe from the control unit has a minimum propagation time from the leading edge of the clock of 15 nanoseconds to a maximum of 665 nanoseconds. Because of this wide variation, a worst case design cannot guarantee that the control unit's Write strobe can always occur in



tc IS THE CLOCK WORST CASE VARIATIONS

tWD IS THE WORST CASE DELAY TIME OF THE CONTROL UNIT

tW IS THE WRITE STROBE PULSE WIDTH FOR MAXIMUM CONDITIONS

Figure 39. A Timing Diagram of a Processor Write Cycle

the proper timing sequence as required by the MF 7112. Therefore, the Write strobe must be stored by a flip flop and clocked with the trailing edge of the master clock. This implementation will delay the actual memory Write time by one (1) microsecond, but will always guarantee the MF 7112 Write strobe within the time frame required. Because the next address change can occur at a worst case of 2 microseconds after the Write strobe, see Figure 39, the delay of one (1) microsecond caused by the retiming is not harmful.

It is suggested that the manufacturer's data sheet for the MF 7112 be reviewed to establish the sequence of the memory Write strobe with resepct to the Cenable and Read clocks previously developed. This Write strobe has already been placed in its proper perspective by the timing logic. The Write timing logic can be expressed as

$$j = t_W \cdot \overline{y}_3 \tag{27}$$

$$k = \overline{t}_W \cdot y_3 \tag{28}$$

$$\text{Memory Write Strobe} = y_3 \cdot A_7(P) \quad . \tag{29}$$

$A_7(P)$ is the Write pulse developed in the ring counter design.

Implementing the Write Logic

Equations 23 and 24 were expressed in terms of a j-k flip flop input. This expression was useful as it was predetermined that the Write strobe required resynchronizing due to processor delays as explained earlier in this section.

Implementing the logic equations using standard TTL circuits is shown in Figure 40.

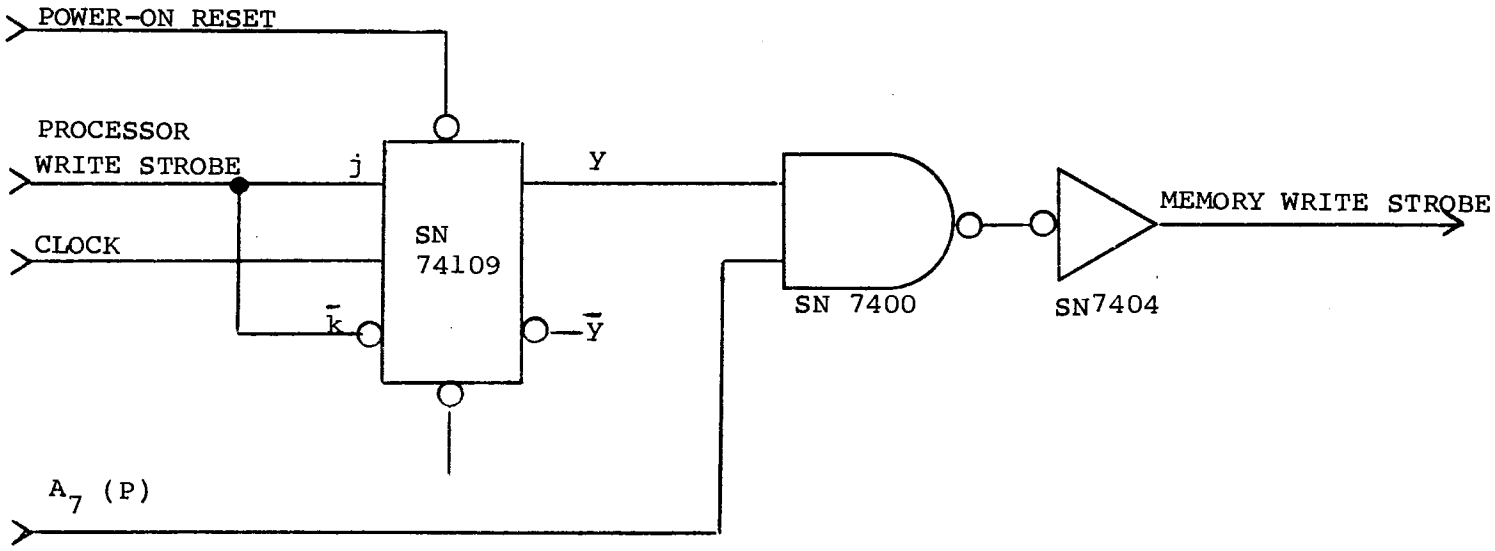


Figure 40. A Schematic of the Write Interface Logic

A parity inhibit signal is also included. This will be shown as a requirement in Chapter V. For now this signal will be expressed as

$$\text{Parity Inhibit (PI)} = \overline{y}_3 \quad . \tag{30}$$

CHAPTER IV

THE READ ONLY RANDOM ACCESS MEMORY

Introduction

General Uses

Small machine uses in which processors are applied to applications such as Point of Sale terminals, Interaction Cathode Ray Tube Displays, Bank Deposit terminals, and numeric control, often require a Read Only Random Access memory (ROM) to compliment the Random Access Read/Write memory. The amount of ROM varies with the type of application. In the case of the so-called "Smart Terminal", which is largely customer programmable, the requirement for ROM is often small. On the other hand, in some applications in which the terminal is factory programmed, large amounts of ROM are used which in turn requires only small amounts of Read/Write memory. In either case, the design principles presented here are suitable for all types of applications.

The discussions following in this section center on the design criteria required to mix ROM and Read/Write memories on the same data and address bus. The discussions are divided into three parts, a general ROM architecture, followed by a survey of commonly available semiconductor ROM's, and concluding with data and address bus precautions including a switching speed design analysis. Memory printed circuit decoding logic is introduced in Chapter V. The appendix will contain the final

schematic of the ROM printed circuit card.

The Read Only Memory Architecture

Problem Definition

For the purpose of this discussion the memory size will be assumed the same as previously outlined, 6144 words of ROM, 16 bits per word. Parity will not be included for reasons outlined in Chapter V. The ROM will be physically located on one printed circuit card. This is primarily due to the number of memory words necessary in the ROM. Smaller amounts of memory will require less printed circuit card space. It is often common in very small machines for the ROM to share the same printed circuit card with the Read/Write memory.

Bus Interconnections

For paralleled ROM and Read/Write memories on separate printed circuit cards the data and address connections are done in the least complex manner by making all interconnections on the back panel. This allows the simplest field replaceable concept and makes production testing easier and less costly. In the case of paralleled Read/Write and Read Only memory, the ROM and the Read/Write memory share the same address bus and the same data bus. A block diagram of the ROM printed circuit card is shown in Figure 41.

Chip Select Decode Implementation

The conventional semiconductor ROM now being offered by industry has standardized on two methods of selecting individual memory chips. For memory elements less than 256 words, the select input is usually directly

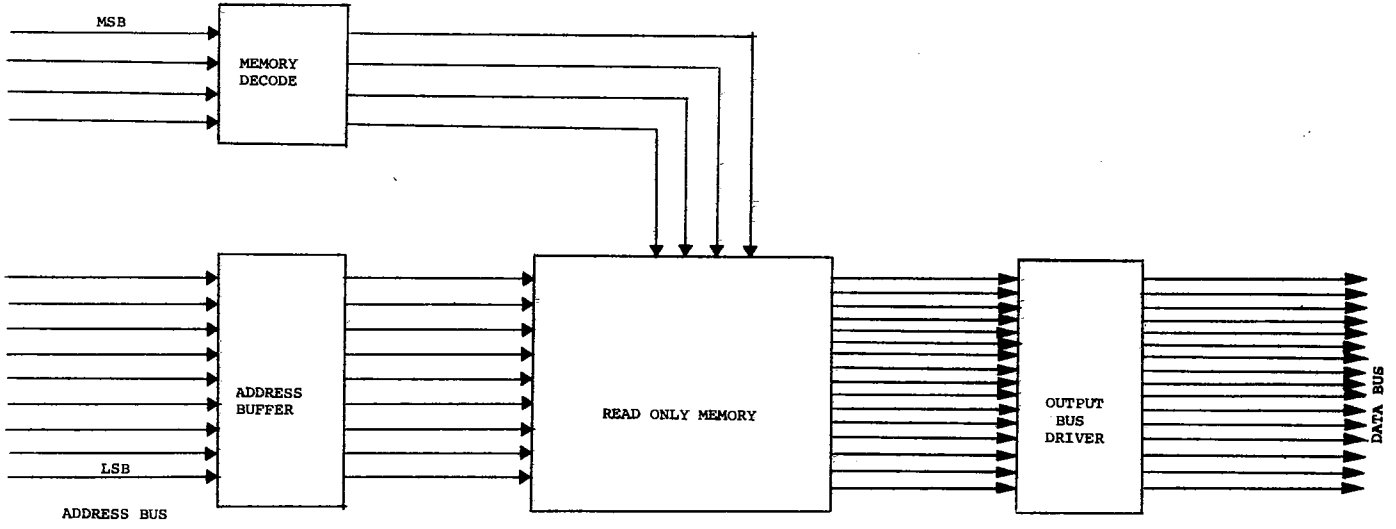


Figure 41. A Block Diagram of the ROM Printed Circuit Card

connected to the elements output transistors. This method is used in lieu of internal element decoding and permits the WIRED OR connections for memory expansion but requires additional decode logic on the printed circuit card. Decode methods are described in Chapter V. For memory elements larger than 512 words, the memory element generally includes internal decoding that causes an individual memory package to disconnect from the data bus. This disconnect permits the WIRED OR condition allowing other memory elements to connect to the bus without interference. The size of the internal decode is limited by package pin connections. The Fairchild Semiconductor, Inc. MOS, 512 X 8, ROM chosen for the example has internal decode permitting memory expansion to 16,384 words with four input pins devoted to select decoding. The user would specify the desired select coding at the time the memory was coded by the semiconductor manufacturer.

Output Data Bus Buffering

The two semiconductor technologies that are most common to ROM, bipolar and MOS, require individual consideration when the connection to the data bus is being implemented. The reader's attention is directed to the last topic of this chapter for output switching time considerations.

Both the bipolar and MOS technology ROM's are implemented with the output buffering either tri-state or open node, i.e., open collector or open drain. Tri-state is the fastest switching time configuration but is limited in the number of output nodes that can be connected together due to leakage currents adding to a limit exceeding the receiving elements capacity. Also, tri-state logic adds current spikes to the data bus which for some applications are intolerable. On the other hand, the open

node connection can be expanded to large memory capacities but is the slowest method for switching time. The designer must choose the output buffering that best meets the application.

In order to meet the access time requirements of the processor, it has been the author's experience that MOS outputs require buffering by a device capable of large capacitance drive such as TTL devices.

Input Node Buffering

As in Output Data Buffering, the two semiconductor technologies must be considered individually. However, for input nodes the reason for individual consideration differs greatly.

The bipolar TTL technology is a current sinking family and when many inputs are combined, such as found in a memory address configuration, this combined input current becomes very large. For instance, assume a memory whose size is 6,144 words and is made up from 256 word ROM's. If the memory is 16 bits wide then the number of input nodes is 96. Multiplying 96 by the conventional 1.6 milliamperes per TTL input, and the designer finds he must supply a driving device capable of sinking 153.6 milliamperes and maintain a logic low level net exceeding the TTL low voltage of 0.4 volts. The value of current changes for the bipolar ECL family and the current must be sourced instead of sinking, but the same rules apply and input loading becomes an important consideration. One method of providing the necessary drive without adding excessive delay is to parallel TTL gates or buffers such as the SN 7400 gate or the SN 7440 buffer. This arrangement provides minimum propagation delay and adequate current drive. However, the memory cost is increased by the additional address drivers, and the printed circuit card area allowable for memory

is decreased.

The MOS technology has an input node drive problem that differs considerably from the bipolar technology. In the case of MOS Read Only Memory, most applications are TTL logic driving MOS memories. Even though the manufacturer's data sheet claims TTL compatibility, a careful review of the device input specifications generally reveals that the MOS memory is close to being compatible but for worst case design considerations will not accept TTL output levels and guarantee operation. Therefore, the designer must make provisions to guarantee operation. Specifically, the problem area is in the TTL output node being guaranteed to be not less than 2.4 volts, while the MOS device is specified at a guarantee level of, V_{ss} - 1 volt. The power supply, V_{ss}, is 5.0 volts for TTL operation; therefore, the guaranteed high MOS input level that can be tolerated is 4.0 volts which is 1.6 volts difference from the TTL output guarantee of 2.4 volts. One of the best methods of overcoming this incompatibility is to add a pull-up register to the TTL output node. This addition will guarantee a high input level of 5.0 volts with a penalty of added components and extra current used by the pull-up resistor. The resistance value is selected as a function of the memory speed and the power supply current available.

For memory applications where the MOS inputs are operated in fast cycle times, fast being defined as cycle times and address changes less than 300 nanoseconds, the designer must contend with the average power dissipation in the TTL driver. The MOS address inputs when connected together to form a large memory block becomes a substantial capacitance to drive and increases the power dissipation of the input driver. For such applications, the SN 75361 is suggested as a possible driver meeting

all necessary input drive requirements if an external pull-up resistor is used.

Survey of Semiconductor ROM

General

The majority of semiconductor ROM's available are divided into two categories, bipolar and MOS. As a general rule, the bipolar ROM's are small capacity but with very fast access times. The cost of bipolar ROM's is generally 2.5 times that of a MOS ROM. The bipolar ROM is often found in small capacity, high speed applications such as microprogramming or cache memory. The MOS ROM is most often found in the program portion of memory where large capacity is required but access time is not critical. For these reasons this design will restrict itself to the MOS ROM's which fit the application better than a bipolar ROM. However, for reference, the bipolar ROM's most commonly used are listed in Table XII. The MOS ROM's offered as catalog devices are listed in Table XIII.

Data Bus Switching Considerations

The Need for a Bus Driver

Most manufacturer's data sheet for semiconductor ROM's specify the ROM outputs as designed to be WIRED OR (or WIRED AND) with other memory devices connected to a common data bus. Even though this type operation is functionally satisfactory it is quite often impossible to drive a data bus due to the associated node capacitance not normally accounted for on a manufacturer's data sheet. The problem can be defined as a switching time consideration in that the standard ROM output transistor cannot

TABLE XII

A TABLE OF BIPOLAR READ ONLY SEMICONDUCTOR RANDOM ACCESS MEMORIES

MANUFACTURER	ROM CATALOG NUMBER	MEMORY SIZE (BITS)	MEMORY ORGANIZATION	PACKAGE DIP	AVAILABLE AS A PROM	
					YES	NO
TI, FSC, MM, NAT'L, H, MOT, SIG	SN 74187	1024	256 X 4	I6 Pin	X	
TI, MM NAT'L, MOT H, SIG	SN 74186	412	64 X 8	16 Pin	X	
TI, MM NAT'L, MOT H, SIG	SN 74188	256	32 X 8	16 Pin	X	
SIG, NAT'L	8228	4096	512 X 8	24 PIN		X
MM, INTEL	MM5240	4096	1024 X 4	24 Pin	X	
MM	MM5205	2048	256 X 8	24 Pin	X	
MM	MM5280	8192	1024 X 8	24 Pin		X
MM	MM5290	10,368	1152 X 9	24 Pin		X

TABLE XIII

A TABLE OF MOS READ ONLY SEMICONDUCTOR RANDOM ACCESS MEMORIES

MANUFACTURER	MEMORY SIZE (BITS)	MEMORY ORGANIZATION	CATALOG NUMBER	PACKAGE SIZE (DIP)	TECH-NOLOGY	ON CHIP MEMORY SELECT DECODING	
						YES	NO
Motorola , RCA	1024	256 X 4	MCM 14524	15	CMOS		X
Motorola , T.I.	2560	256 X 10	MCM 1150	24	P		X
Motorola , T.I.	4096	512 X 8	MCM 140	24	P		X
Nat'l. FSC MOSTEK	4096	512 X 8	3514	24	P	X	
Motorola	8192	1024 X 8	MCM 560	24	n		X
Intel	2048	256 X 8	1702 A(PROM)	24	P		X
AMI	8192	2048 X 4	58865		P		

drive large capacitance such as accumulates on a data bus. The designer must decide if the ROM output transistors need to be buffered by another device.

Analysis

The first step in the design analysis is to define the problem. A typical bus structure is shown in Figure 42. A number of ROM outputs are connected in parallel, each output being tri-state, i.e., a logical 1, a logical 0, and a high impedance output off state. Only one line of the 16 line data bus is shown in Figure 42 for simplicity.

As shown in Figure 42, the amount of capacitance that each output must drive increases as a function of the printed circuit card memory size, i.e., the number of ROM output nodes connected together. The total capacitance as seen by the ROM output transistor is summed in Table XIV. The typical ROM output consists of a push-pull connection. The majority of these outputs are TTL compatible unless they were made for a specific application. Figure 43 represents a typical push-pull MOS driver.

The ROM chosen for this example is a Fairchild Semiconductor, Inc., 3514. The 3514 is specified to be TTL compatible and have a guaranteed maximum output capacitance of 12 picofarads (pf). The designer must know the drain to source resistance of transistors Q_1 and Q_2 (shown in Figure 45) in order to calculate the switching speed of the transistor. However, the true non-linear drain to source MOS impedance is not available on the manufacturer's data sheet. The only information that is available for the designer is the saturation resistance of the two transistors. Often this is discussed in the D.C. specification of the device data sheet. For transistor Q_2, the 3514 is specified as 2 milliamperes

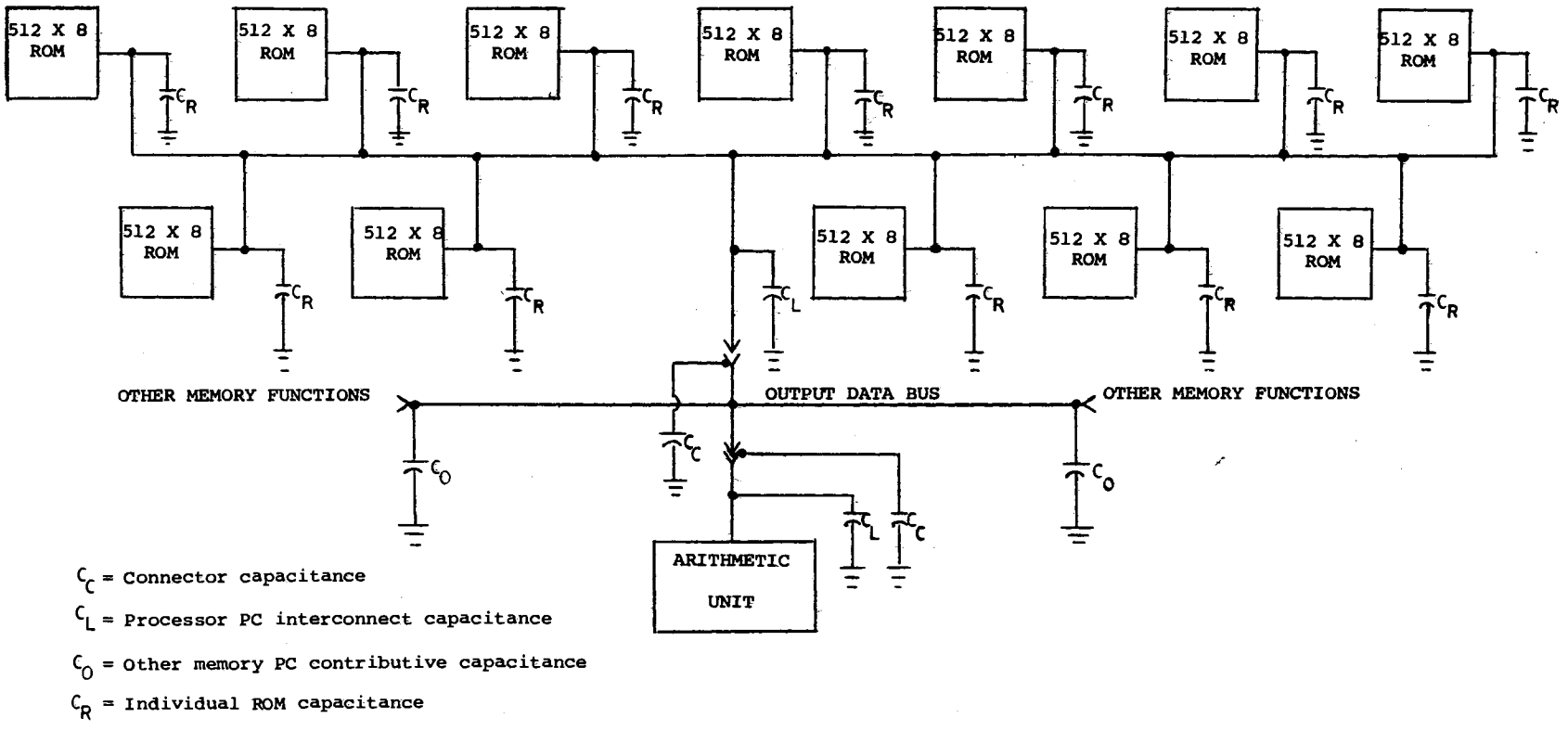


Figure 42. A Pictorial Representation of a Typical Memory Bus Structure

TABLE XIV

SUMMATION OF THE ACCUMULATIVE BUS AND NODAL CAPACITANCE

CONTRIBUTIVE CAPACITANCE	SYMBOL	CAPACITIVE IN 10^{-12} FARADS	QUANTITY MULTIPLIER	SUBTOTAL SUMMATION in 10^{-12} FARADS
ROM IC	C_R	12	12	144
PC Interconnecting Runs (ROM)	C	1/inch	10	10
PC Edge Connector	C_c	4	3	12
Backpanel Wiring	C_o	1/inch	8	8
PC Interconnecting Runs (R/W)	C_o	1/inch	20	20
2nd PC Card ROM Outputs	C_R	12	12	144
Arithmetic Unit PC Card	C_A	1/inch	10	10
Input Data Register	C_A	8/Input	2 Inputs	16
			TOTAL	354

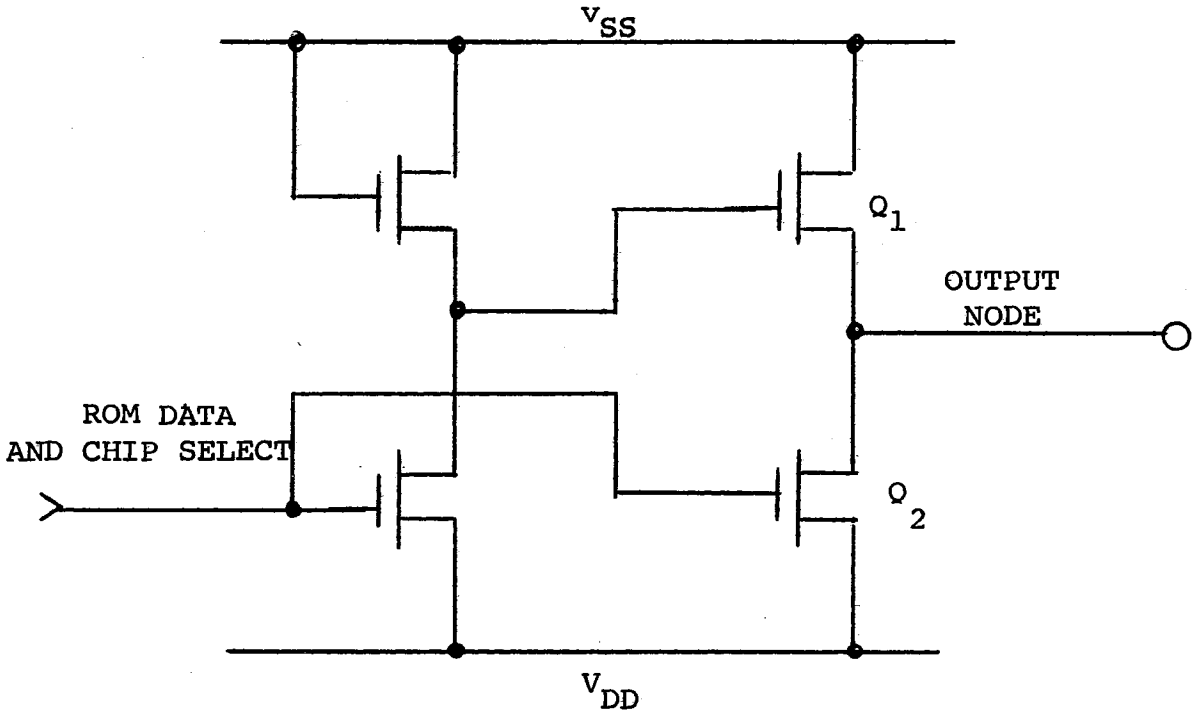


Figure 43. A Typical MOS Output Driver for a ROM

current sink, i.e., current into the output device, guaranteed saturation voltage not to exceed 0.45 volts. Therefore, the saturation resistance can be expressed as

$$R_{SAT}(Q_2) = \frac{0.45}{2 \times 10^{-3}} = 225 \text{ ohms} \quad . \quad (31)$$

The pull-up transistor Q_1 is specified as supplying 100 microamperes, i.e., current from the output node, at a saturation voltage no less than 2.4 volts when V_{SS} equals 4.75 volts. Therefore, the saturation resistance of Q_1 can be expressed as

$$R_{SAT}(Q_1) = \frac{4.75 - 2.4}{100 \times 10^{-6}} = 2350 \text{ ohms} \quad . \quad (32)$$

The MOS transistor resistance not being linear can be approximated for switching time considerations by assuming linear operation of both the pull-up and the pull-down transistor. With a resistance value for the ROM driver transistors calculated, the switching time of the output transistors can be found. The first step is to sum the accumulative node capacitance shown in Figure 42. Table XIV is a summation of this accumulated capacitance which includes all nodes that the output transistors are required to drive.

The initial memory design criteria specifies a memory logic 0 level as being V_{SS} minus 1.0 volt, and a logic 1 level as V_{DD} plus 0.4 volts. An examination of the manufacturer's ROM data sheet indicates that the output switching time is specified at V_{DD} plus 1.5 volts, for both a logic 0 and a logic 1. It is immediately clear that the switching time of the ROM will be increased for the logic 0 case. The method of testing and the application differs greatly. Figure 44 represents a propagation delay timing diagram of the Fairchild 3514 ROM.

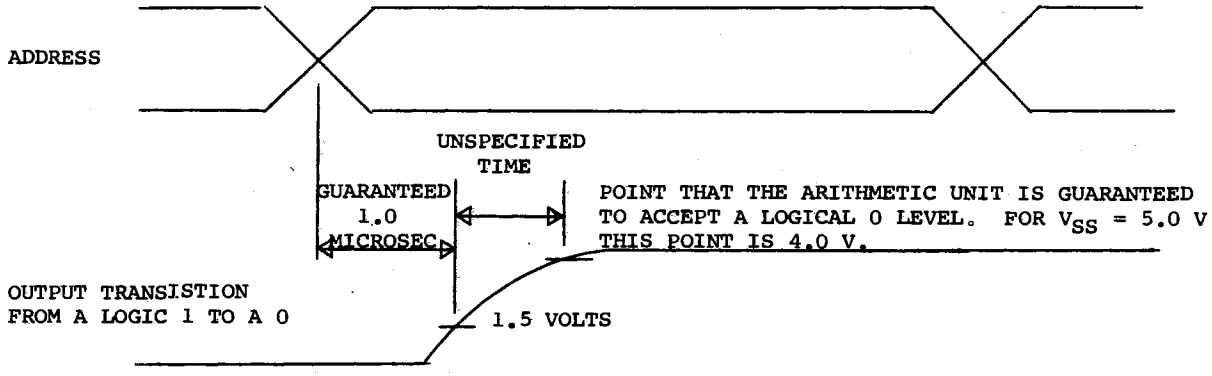


Figure 44. Waveform Diagram Representing the Access Time of a ROM Including Accumulative Data Bus Switching Time

Referring to Figure 44, the equation for the rise time of the output transistor Q_1 can be expressed as a voltage across the total summation of nodal capacitance. The node capacitance is summed in Table XIV. The expression for the voltage is

$$V_c = V_{co} + (E - V_{co})(1 - e^{-t/RC}) \quad . \tag{33}$$

The time of interest is the time required for the output voltage to make the transition from 1.5 volts to 4.0 volts. Using the value of R equal to 2350 ohms, and the summation of the bus capacitance of 354 picofarads, Equation 27 reduces to the solution of time t as

$$\frac{t}{RC} = 1.238 \tag{34}$$

or

$$t = 1.02 \times 10^{-6} \text{ seconds} \quad . \tag{35}$$

The time for the output to transition from a logic 1 to a logic 0 is the access time of the ROM, one (1) microsecond, with the transition time added to the access time to become the total access time of the memory. This total access time is expressed in Equation 36 as

$$t_{access} + t_{transition} = 1.0 + 1.02 = 2.02 \text{ microseconds} \quad . \tag{36}$$

Therefore, the total memory access time from address change until the memory output level reaches the guaranteed logic 0 level is 2.02 microseconds, an increase of more than 100 percent over the original memory element's guarantee access time as specified by the manufacturer.

Based on this increase in access time, one can conclude that the memory data output bus must be buffered when a TTL compatible ROM is

driving a high capacitance MOS bus. This also holds true when the bus node capacitance becomes large due to the summation of node capacitance no matter what type technology is driving another technology. The buffering can be accomplished with either TTL tri-state or open collector drivers. Either is suitable with the tri-state devices exhibiting the fastest switching times. One can also conclude that the tri-state MOS output arrangement is limited in use to very small memory applications before switching time becomes prohibitive.

CHAPTER V

SUPPORT LOGIC

Parity

Introduction

Parity is often used as a simple way for detecting and checking errors. By adding a check bit P to the memory matrix an error checking circuit can be used to detect circuit malfunctions.

For the small machine applications odd or even is generally used, opposed to more complicated methods. Odd parity is ordinarily used for two reasons: (1) the zero digit is often represented by loss of signal; and (2) the odd parity used in an even number of bit words automatically detects an error in the common all "1's" or all "0's" condition that most often occur in circuit malfunctions.

Odd parity is normally more advantageous in a dynamic memory because the memory will assume a logic level of zero (negative logic) when most circuit malfunctions occur. This, in general, is a true statement for malfunctions such as loss of the refresh signal, loss of a clock, or memory power. This is not to infer that all failures cause a zero level, but only that a majority of failures will.

Parity is more often checked on the Read/Write portion of memory only, rather than expanding to the Read Only portion. This is generally an economic consideration, as the extra parity bit required to check the

ROM is an added expense that cannot be justified by the normal amount of ROM failures. On the other hand, error checking of the Read/Write portion is almost an essential requirement.

For a dynamic memory, the parity section consists of:

1) parity generation;
2) parity bit storage;
3) parity check;
4) parity error detect; and
5) parity strobe.

Parity generation for odd parity is expressed in general terms as:

$$X_3 \oplus X_2 \oplus X_1 \oplus P = 1 \quad . \tag{37}$$

This, of course, is also true for the parity check.

Another desirable feature is to allow the customer to add to memory in the building block method. Often the added memory required will be one or two thousand bytes of memory rather than words. Therefore, the parity generation and check portions must be able to account for "missing" parts of memory without some manual intervention of the operation.

Basic Design

The first step will be to block out what is to be accomplished. Figure 45 is a block diagram of the parity circuit.

The requirement for odd parity generation is expressed as:

$$DR_1 \oplus DR_2 + DR_3 \oplus \cdots \oplus DR_{16} \oplus p = 1 \tag{38}$$

This equation is satisfied by the 8262 parity integrated circuit and

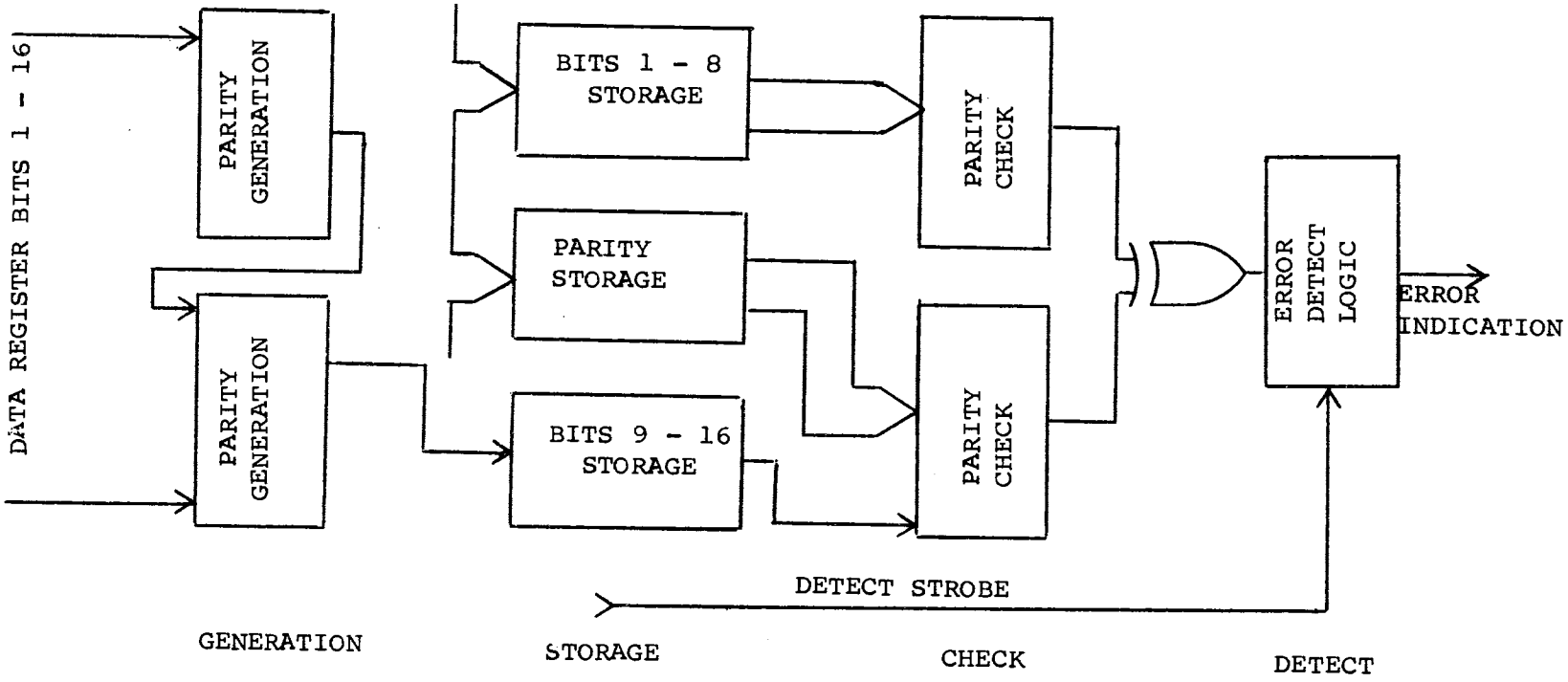


Figure 45. A Block Diagram of a 16 Bit Memory Parity Circuit

is shown schematically in Figure 46. For a byte to be implemented the parity generation is expressed as:

$$DR_9 \oplus DR_{10} \oplus \cdots \oplus DR_{16} + p = 1 \quad . \qquad (39)$$

Because of the relatively long propagation delay times of the 8262 and the fact that the data is true for 500 ns, the two parity check 8262's will be externally exclusive ORed together using another module of the SN 7486 gate originally used in the parity generation. One of the inputs to the left half of memory will be grounded. The simplest method for compensating for a byte of memory instead of a word is to not use any complicated logic to disable the 8262 for the left side of memory, but rather, make use of the fact that when memory is missing (physically removed) the data output bits are all logic 1's by virtue of the pull down resistors of the sense amplifier. However, one must use the even output node of the 8262 for the left half of memory. This byte inhibit is shown in Figure 47.

Refresh Inhibit

One more item must be accounted for. When more than one RAM output is connected in a wired OR condition, during refresh time the outputs of the RAM's of each data bit, including parity, are invalid. This is because the low logic 0 state is predominate and a logical 0 of one portion of memory may be the correct bit; however, a logical 1 of another portion may be correct, but the wired OR output will assume a logical 0 and cause parity error to occur in the check. Therefore, a means to inhibit parity during refresh is required. Figure 48 represents the final configuration of the parity check circuit. Notice that the parity detect flip flop is

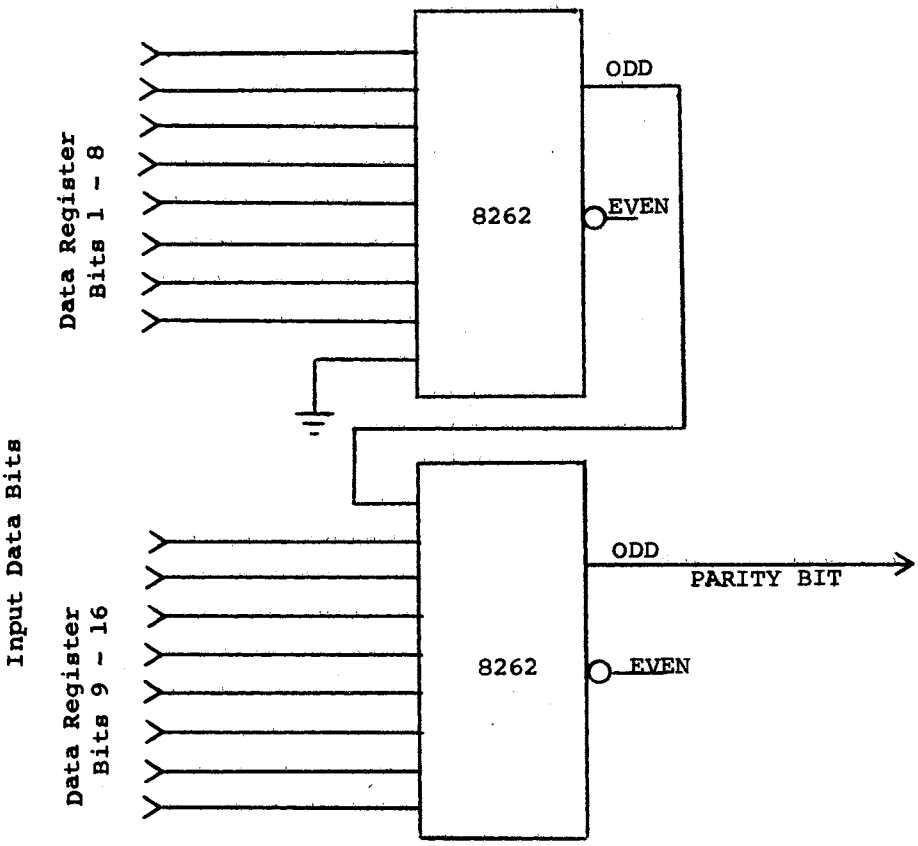


Figure 46. A Schematic of the Odd Parity Generation

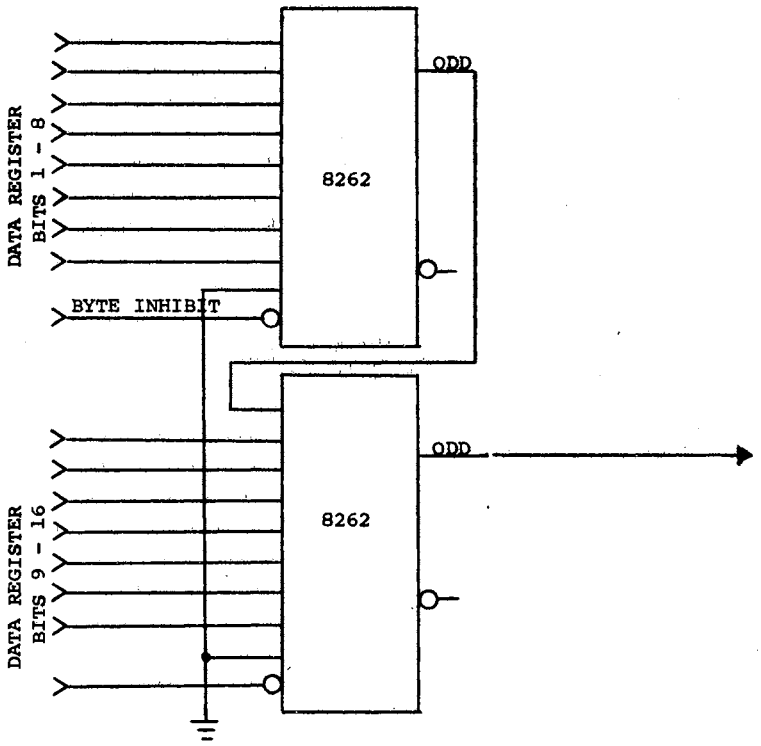


Figure 47. A Schematic of Odd Parity Generation Including a Byte Inhibit

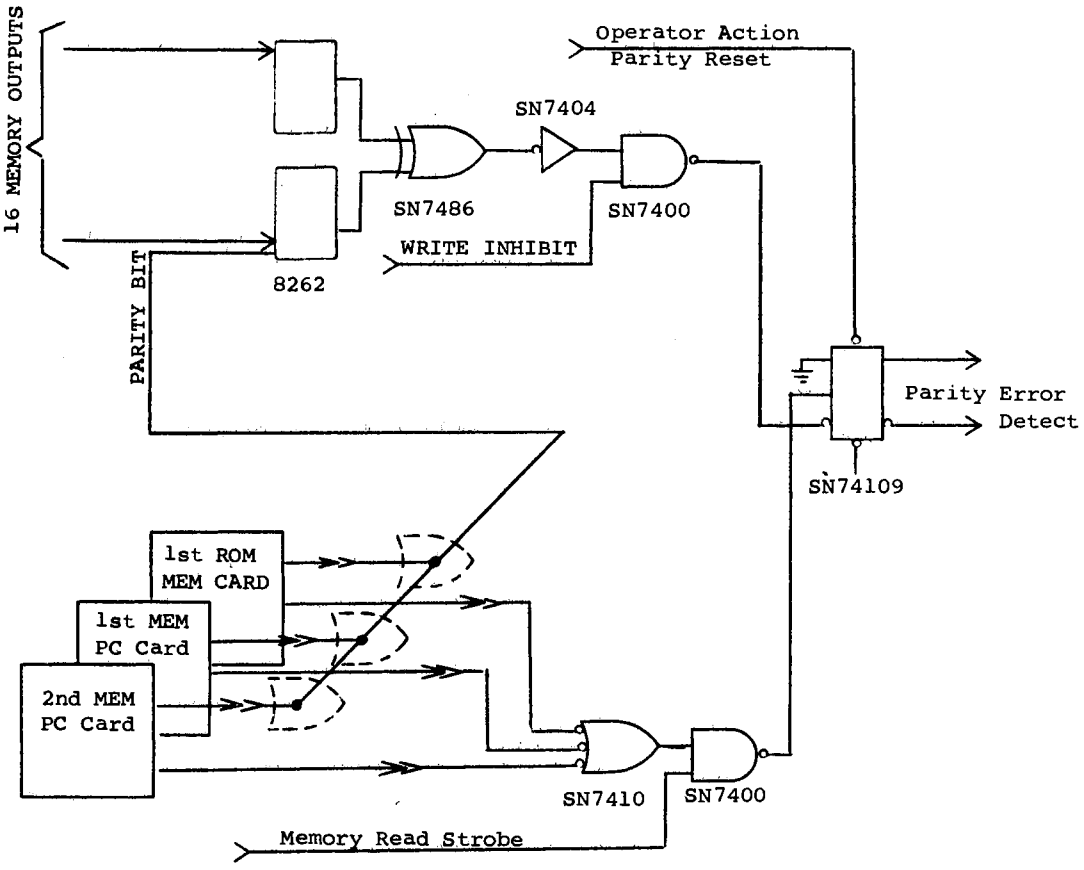


Figure 48. A Schematic of the Odd Parity Check Including the Parity Error Detect Logic

a D clocked flip flop (a SN 74109) with the set input connected to ground. The flip flop is strobed by the read strobe developed in Figure 16.

The circuit is designed in this manner to force some type of system or operator action to reset a parity error indication. With the set side of the D-type flip flop grounded an error will permanently lock the flip flop into a parity error indication until the pre-set input of the flip flop is grounded or power is turned off.

Test and Diagnostics

Introduction

Like their large machine counterparts, a machine normally will have a power turn-on Test and Diagnostic (T & D) routine written into the Read Only Memory. This routine can be as complex or as simple as the application requires. One of the more popular simple and inexpensive routines will first use a small portion of the Read Only Memory to test the Read/Write meory, transfer the routine from ROM to Read/Write and then test the full ROM complement, and finally run a test of the communication circuitry if an off line capability exists. The testing can be extended to the other peripheral devices if used.

The discussion of the above areas will be restricted to the memory as other testing is beyond the scope of this paper. Let us develop a hypothetical routine to test the memories. The designer can expand or delete tests depending on the application.

Implementation

The T & D starts with a routine written into ROM that is initiated when power is turned on. This routine will generally consist of a Read/Write memory test exercising the memory Read/Write, addressing capabilities, and parity circuit check out. If the parity portion of memory is first checked, parity can then be used to test the remainder of the Read/Write memory instead of requiring additional ROM for a "brute force" compare. In order to check parity, the parity generation circuit developed in the first part of this chapter requires modifying.

With the addition of the parity error lead a parity error can be forced and checked to insure an error was detected, and then force a no error condition and again insure that the no error was detected. The flow chart of Figure 49 represents the sequence of events that are required in the parity error test.

When the last portion of the test represented by the flow chart is reached, the parity section of memory has been reasonably guaranteed to be functional and reliable.

Establishing that the parity portion of memory is reliable gives the designer a powerful tool in checking the remainder of the Read/Write memory. By using parity, the memory can be exercised to insure that all functions are operational. As a minimum, the test should include:

1) memory address exercising;

2) all test altered (Write); and

3) all cells read.

When the Test and Diagnostic has established that the Read/Write memory is operational the test routine can then be transferred from ROM to the Read/Write memory and the same type of test performed to insure

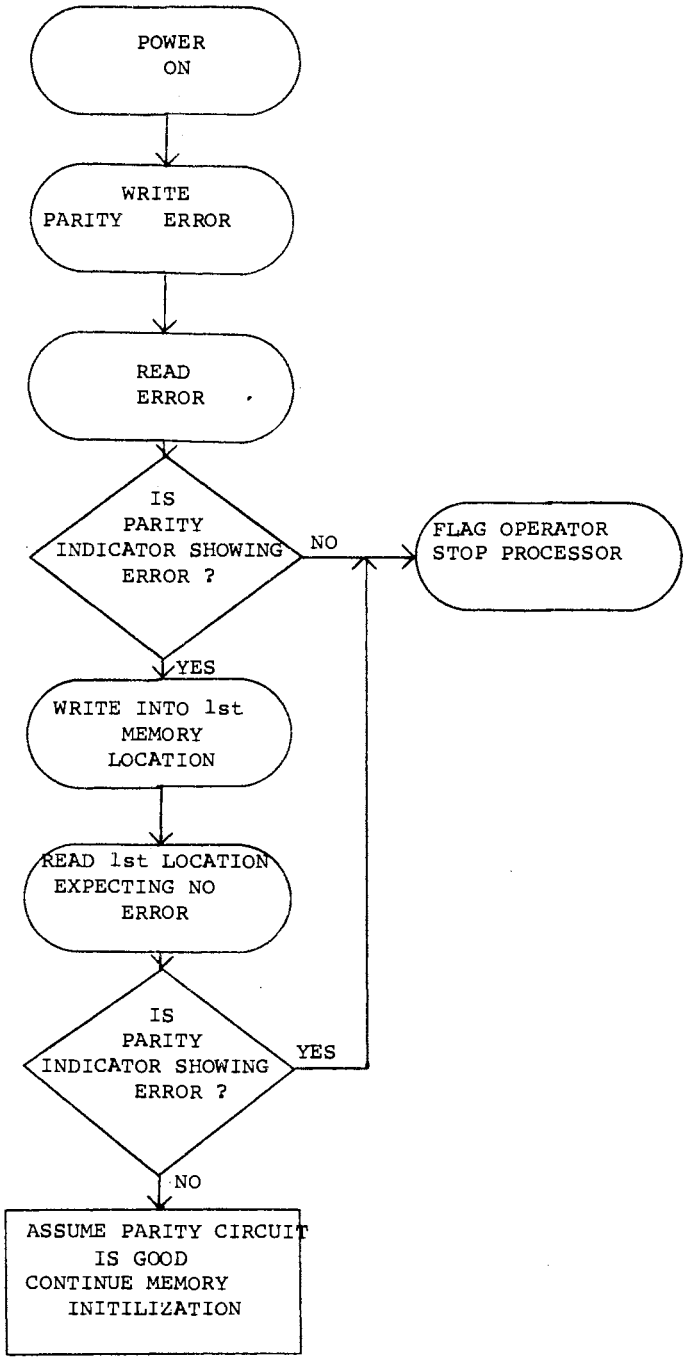


Figure 49. A Flow Chart of the Initialization Test and Diagnostic Routine

that ROM is functional. This involves a check of the ROM's logical 1 outputs. These outputs are summed and compared with a pre-established sum. A more elaborate Teas and Diagnostic routine would be required to establish where the error occurred.

Use of a 8262 parity generator Integrated Circuit connected as shown in Figure 50, allows the implementation of the necessary inhibit by adding one wire. Figure 50 shows schematically this signal lead designated as Force Parity Error. The function of the Force Parity Error signal is expressed as:

$$DR_1 \oplus DR_2 \oplus DR_3 \oplus \cdots \oplus DR_{16} \oplus P \oplus C = 1 \qquad (40)$$

Element C is the parity error signal lead. The signal lead C when forced to a logical 1 causes an error to be detected by the parity check circuit of Figure 50. With this modification to the parity generator circuit, the parity check circuit does not need to be changed.

An extra exclusive-OR gate external to the 8262 is also required to add the Force Parity Error lead. This gate is an SN 7486 quad exclusive OR, TTL integrated circuit.

Memory Decode Logic

Introduction

In the event memory is expanded beyond the limits imposed by the memory elements addressing capability, a method of selecting the specific memory element requested by the control unit must be implemented. The discussion in the following paragraph details one method of decoding the address register information into printed circuit card selection and memory element selection.

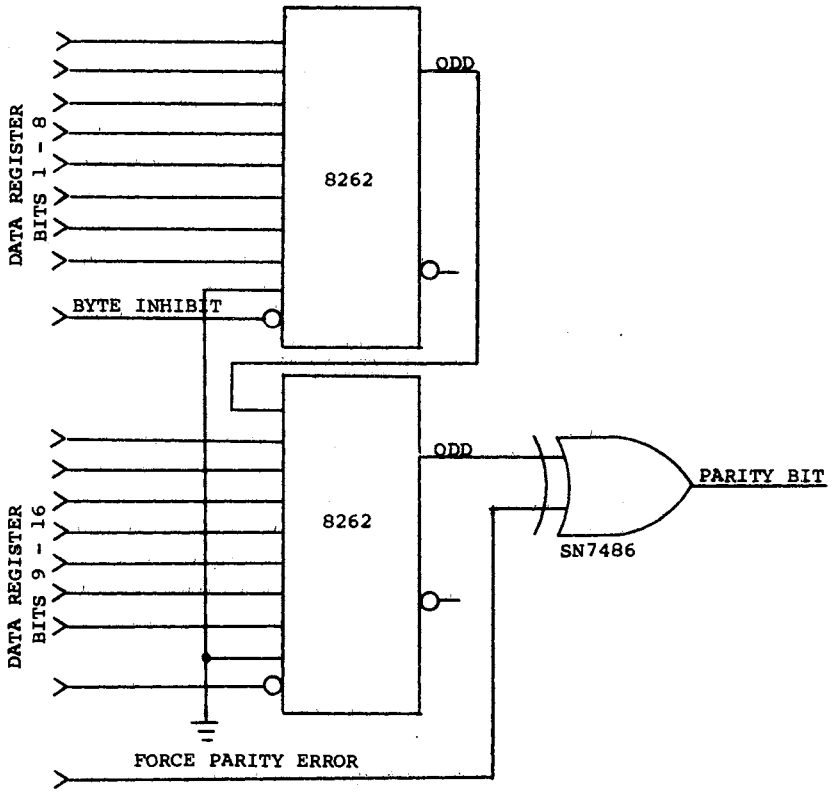


Figure 50. Implementation of the Test and Diagnostics Routine Into Parity Generation

Implementation

This designer of memory systems usually is confronted with three guide lines in developing a memory on a printed circuit card:

1) the design must operate in any memory chassis location assigned with out on-the-card modification;

2) the design must be field interchangeable without the field engineer or technician needing to perform on-site modification to the memory card; and

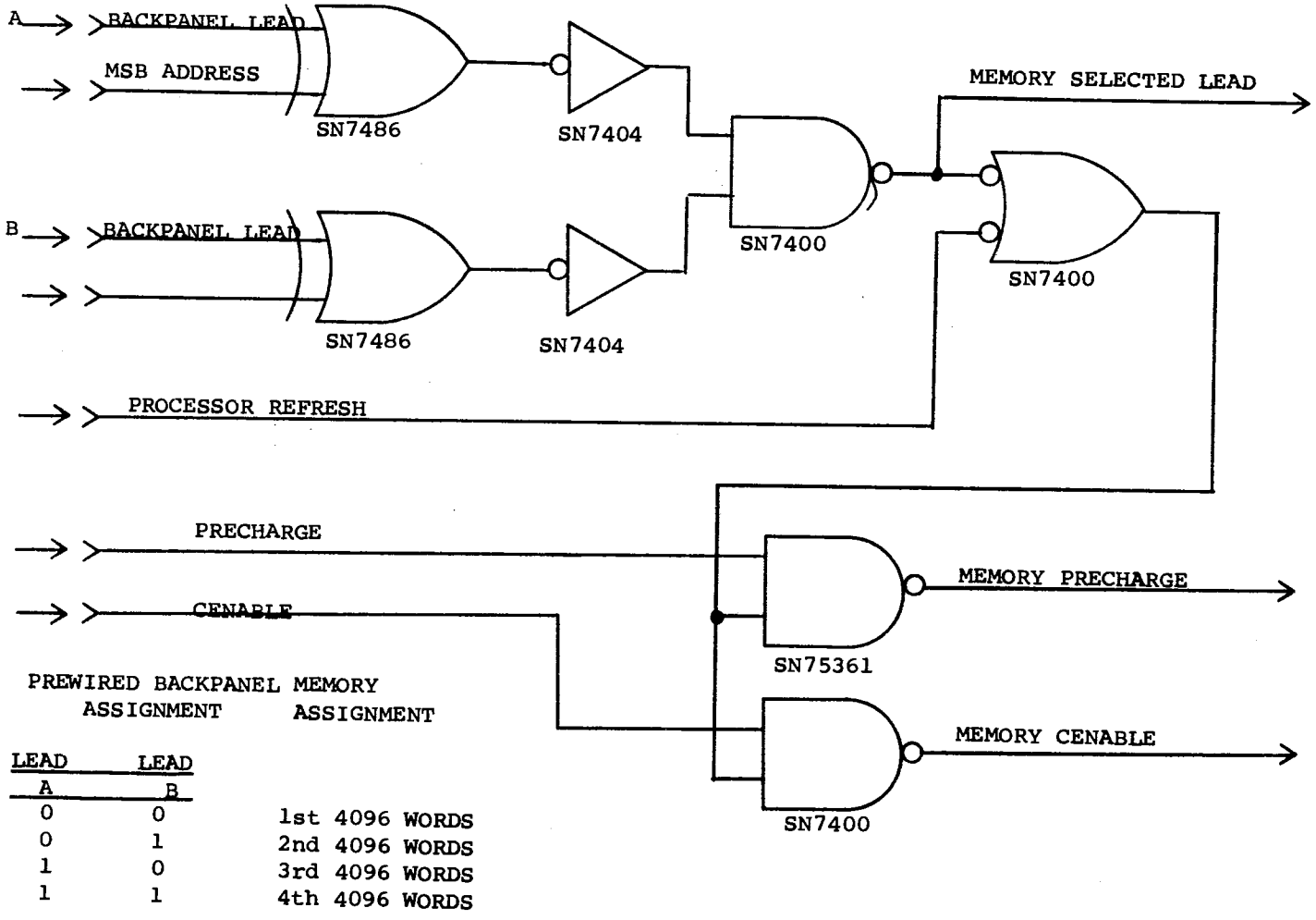
3) the design must operate in any processor memory address location without modifying the printed circuit card to taylor each card for a unique address.

In addition, the decode logic is generally in the section of logic that supplies the signals such as the BYTE INHIBIT signal in this parity discussion.

All of these restrictions can be met by assigning memory on the chassis backpanel. In addition to meeting these restrictions, backpanel selection allows one unique memory design to operate in a multitude of memory applications which in turn decreases the manufacturing costs of the printed circuit card. One extra restriction that must be placed on use of the printed circuit card using this type of design is that the memory must be added consecutively.

The implementation of the memory decode requirements for the Read/Write memory printed circuit card is shown in Figure 51. Exclusive OR gates are used to accomplish the backpanel wiring selection.

Adding memory to the printed circuit card requires one more level of decode. For instance, if 8192 words of Read/Write memory were located on one printed circuit card then the decode could be done as shown in



PREWIRED BACKPANEL MEMORY ASSIGNMENT

LEAD A	LEAD B	ASSIGNMENT
0	0	1st 4096 WORDS
0	1	2nd 4096 WORDS
1	0	3rd 4096 WORDS
1	1	4th 4096 WORDS

Figure 51. A Schematic for Memory Decode Logic Implementing 4096 Words

Figure 52.

The decode circuit shown in Figure 52 will also perform for a 16,384 word machine when a 1024 word memory element is used. One extra lead must be brought into the SN 7442 decoder. The lead is MSB-3. This breaks the 16,384 words into 1024 word increments. This decode circuit also works well for ROM that does not have on chip decoding.

Read/Write Memory Storage Register

Implementation

For the small machine applications where the processors data input requires a static level, the dynamic RAM or ROM must have a data register. A parallel in, parallel out register such as the SN 74195, MSI element satisfies the requirement. To parallel memory printed circuit cards on uncommitted TTL output open collector is required. The uncommitted output allows ROM and Read/Write memories to be wired OR on the backpanel. Figure 53 is a schematic of the register and the inverters used to make the data output lines uncommitted. The schematic is made to fit the 16 bit machine.

Also shown are pull up resistors normally required for sense amplifiers. In this example the SH 75108A was assumed to be used for the sense amplifier. The SN 75108A is an uncommitted output high impedance to TTL level device whose output is designed for memory wire - OR. The resistors complete the output circuit. The resistance value should be a function of the designers speed, power requirements and the sense amplifiers low level current sink capabilities.

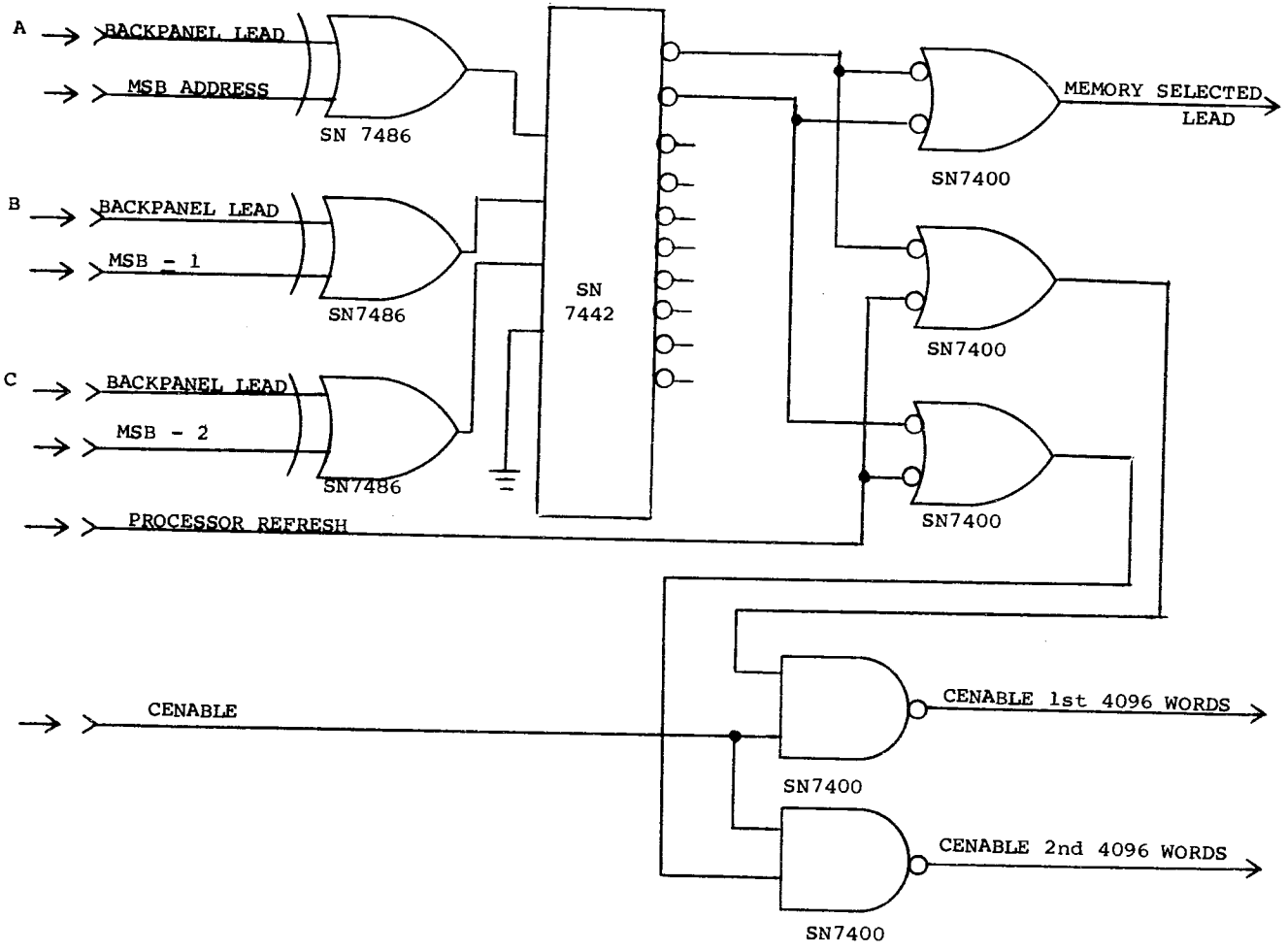


Figure 52. A Schematic for Memory Decode Logic Implementing 8192 Words

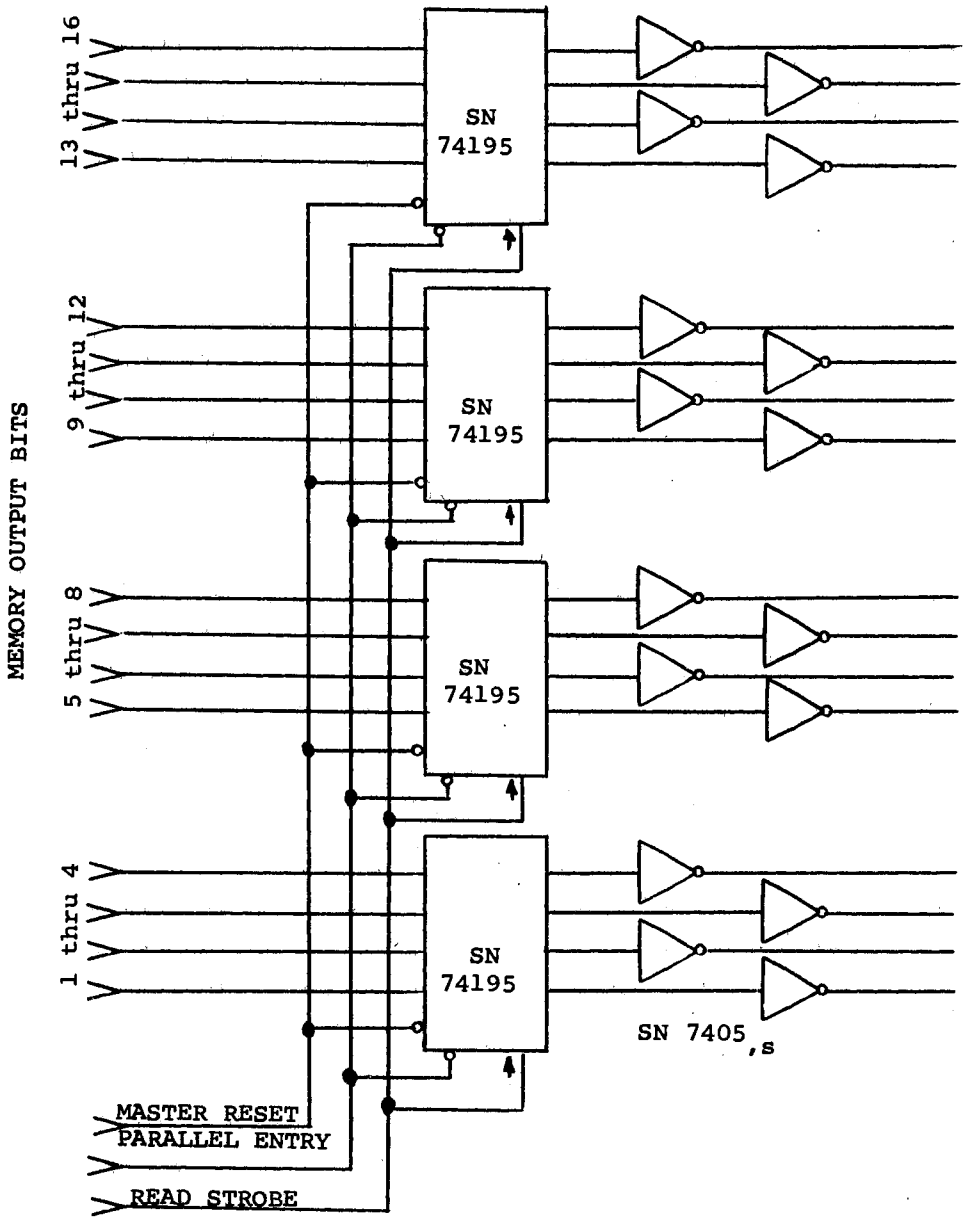


Figure 53. A Schmatic of the Output Data Register Including Open Collector Gates for Memory Bus Wire-OR Connection

CHAPTER VI

SUMMARY AND CONCLUSIONS

Problem Summary

When adapting a semiconductor memory to a Small Machine application, five basic problems must be considered. These problems can be summarized as the following.

1) Choosing the type of memory based on the requirements and cost needs of the machine applications. Two technologies are available, bipolar and MOS. Each technology can be divided into two distinct categories. Bipolar is divided into TTL for medium speed applications, ECL for very high speed application. MOS is divided into static memory for simple but small storage applications, dynamic for complex but dense storage applications.

2) Interfacing between the three main block functions of the processor and memory. Many techniques are available, but two are most common because of their simplicity and cost effectiveness. These techniques are the ring counter and the monostable multivibrator.

3) Paralleling the Read/Write and Read Only memories on the bus. Both the address and data bus require special attention to the timing and loading interface.

4) Adding the memory peripheral logic. This logic requires special attention both because of its technical, i.e., timing, loading and logic states, and the added cost and packaging needs. The general

peripheral logic can be summarized as parity, test and diagnostics, and the data storage register.

5) Proper adaptation of the processor and memory refresh needs. This is applicable to a MOS dynamic memory only. Particular attention must be paid to the processors operating conditions and the needs of the memory.

Conclusions

The trend of the computer industry is away from magnetic core to semiconductor memory. This trend is more dramatic, at present, in small machines rather than the larger computers; however, the large processors are beginning to appear with semiconductor main memory in greater numbers. Soon, all of the main computer memory will be semiconductor as the memory cost is further reduced and the magnetic core can no longer compete. This trend is substantiated by the investments of core memory manufacturers into semiconductor subdivisions. With the advent of the magnetic bubble technique and charge coupled devices, the previous dominated domain of the disk and tape handlers is being invaded with the conclusion inevitable.

The first design step to be taken is to choose the memory that best fits the application. This choice is often a compromise based on a broad need and application requirement. In most cases an adaptation of the memory will be required. For static memories, this adaptation is minor. For dynamic memories, the adaptation can become complex. However, with proper techniques and precautions almost all applications can be satisfied.

The MOS dynamic memory is the most difficult type of memory to adapt

to a processor. However, in a majority of cases, the dynamic MOS memory is the most cost effective of the semiconductor memories, giving the largest storage capacity for the least power consumed, the lowest cost per bit, and the smallest package size.

The proper interface logic is chosen as a function of many variables. The principle variables are cost, performance, packaging requirements, environmental requirements, and power needs.

The final consideration of the designer is the method of refreshing the memory. Many techniques are available, planar refreshing, charge pump, and processor controlled to name a few. All of these methods are related and require a consideration of the processor's operating conditions. With these considerations held prominent, the most efficient technique of refresh and interface logic can be decided.

SELECTED BIBLIOGRAPHY

Chu, Yaohan.

1962 Digital Computer Design Fundamentals. New York: McGraw-Hill.

Crawford, R. H.

1967 MOSFET in Circuit Design. New York: McGraw-Hill.

The Engineering Staff of American Micro-Systems, Inc.

1972 MOS Integrated Circuits. New York: Van Nostrand Rienhold.

The Intel Corporation.

1973 The Intel Memory Design Handbook. Santa Clara, California.

APPENDIX A

A LIST OF SEMICONDUCTOR MEMORY MANUFACTURERS

ADDRESS	ABBREVIATION
American Micro-Systems, Incorporated 3800 Homestead Road Santa Clara, California 95051	AMI
Advanced Memory Systems 1276 Hammerwood Avenue Sunnyvale, California 94086	AMS
Fairchild Semiconductor Corporation 464 Ellis Street Mountain View, California	FSC
General Instrument Corporation Microelectronics Division 600 West John Street Hicksville, New York 11802	GI
Harris Semiconductor P. O. Box 883 Melbourne, Florida 32901	H
Intel Corporation 3065 Bowers Avenue Santa Clara, California 95051	I
Intersil Incorporated 10900 North Tantau Avenue Cupertino, California 95014	INT
Monolithic Memoires, Incorporated 1165 East Arques Avenue Sunnyvale, California 94086	MM
Mostek Corporation 1215 W. Corsby Road Carrollton, Texas 75006	MOS
Motorola Semiconductor Products Incorporated P. O. Box 20924 Phoenix, Arizona 85036	MOT
National Semiconductor Corporation 2900 Semiconductor Drive Santa Clara, California 95951	NATL
Signetis Corporation 811 East Arques Avenue Sunnyvale, California 94086	SIG

ADDRESS	ABBREVIATION
Texas Instruments, Incorporated P. O. Box 5012 Dallas, Texas 75222	TI

APPENDIX B

TWO ALTERNATIVES FOR THE IMPLEMENTATION OF AN ASTABLE MULTIVIBRATOR

There are two basic astable multivibrator designs that are suitable for use in refreshing MOS Read/Write dynamic memories. Each design has its advantage and disadvantages. The first astable has the advantage of small physical size and minimum component cost. The design consists of a monolithic multivibrator with a feedback loop from the output to the input. Figure 54 represents the schematic of the astable using a Fairchild Semiconductor, Inc. 9602 multivibrator. The pulse width of the astable is a function of the internal built in capacitance of the 9602 and external resistor time constant. The period is a function of the external capacitance and resistance time constant. The period can be preselected; however, the pulse width is fixed and can be changed little with the external resistance.

The second astable can be controlled with both pulse widths and period a function of the designer's needs. However, this discrete design has the disadvantage of many components compared with the 9602 design and a large printed circuit card area required to locate these components. Cost of the discrete design is slightly higher than the 9602 monostable. Figure 55 represents the schematic of the discrete design. In either case, discrete or monolithic multivibrator, a pulse width and frequency period variation of 75 percent should be allowed.

The basic pulse width of one-half of the oscillator frequency can be expressed as

$$\tau = C\left(\frac{1}{R_5} + \frac{1}{R_2} + \frac{1}{hie}\right) \quad . \qquad (41)$$

The diodes are placed in series with the transistor's base to protect the transistor base to emitter junction against back biasing, which for a silicon transistor is generally 4 volts.

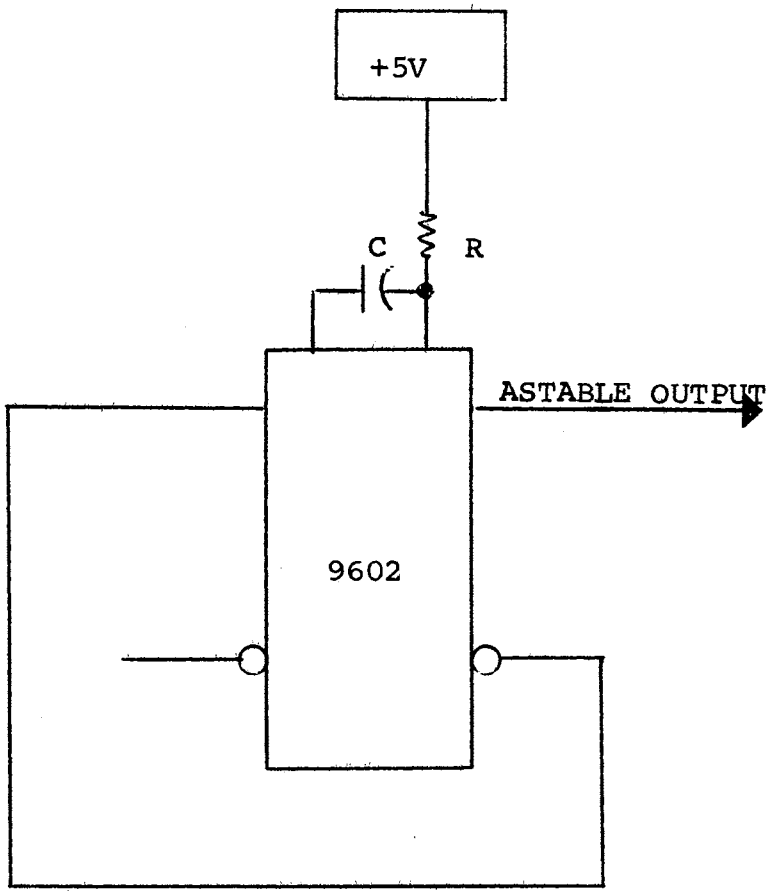


Figure 54. A Schematic of an Astable Using an Integrated Monostable Multivibrator

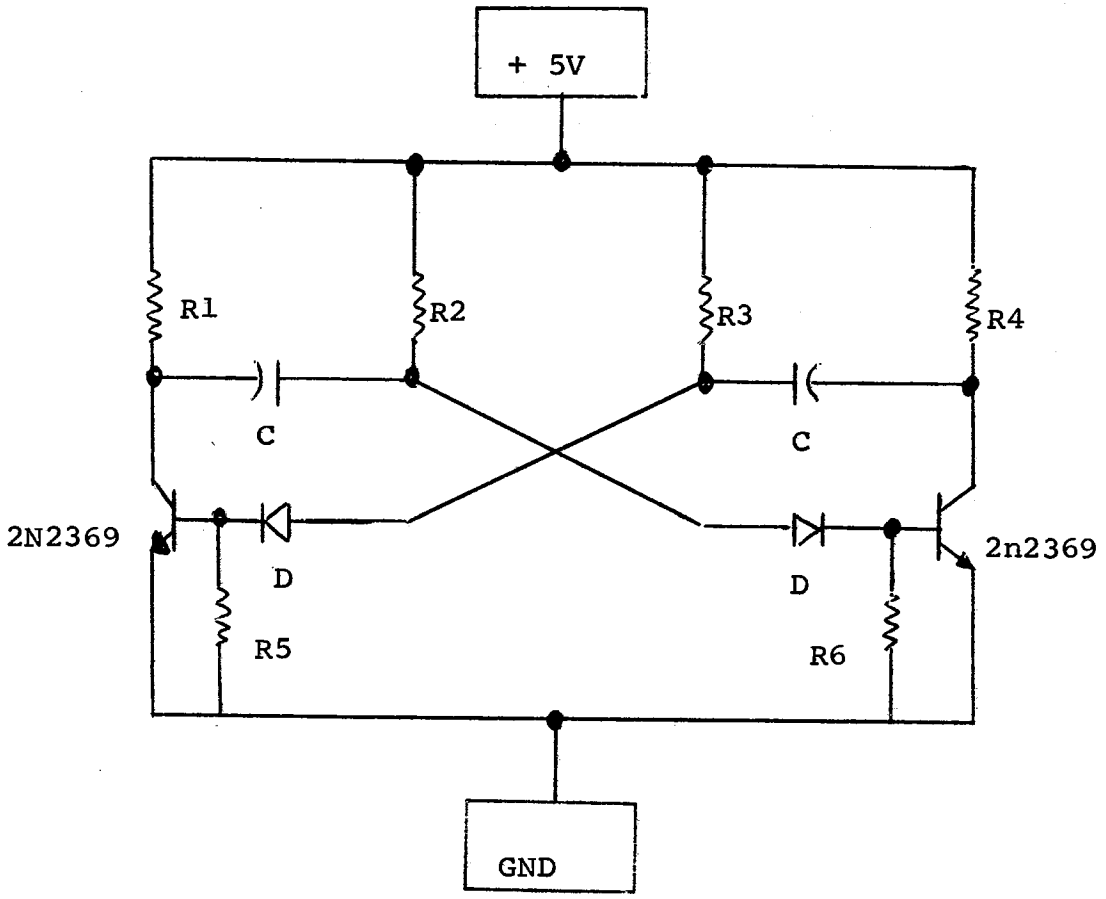


Figure 55. A Schematic of a Discrete Astable

APPENDIX C

A FINAL SCHEMATIC OF A READ/WRITE DESIGN USING MONOSTABLE MULTIVIBRATOR FOR INTERFACE LOGIC

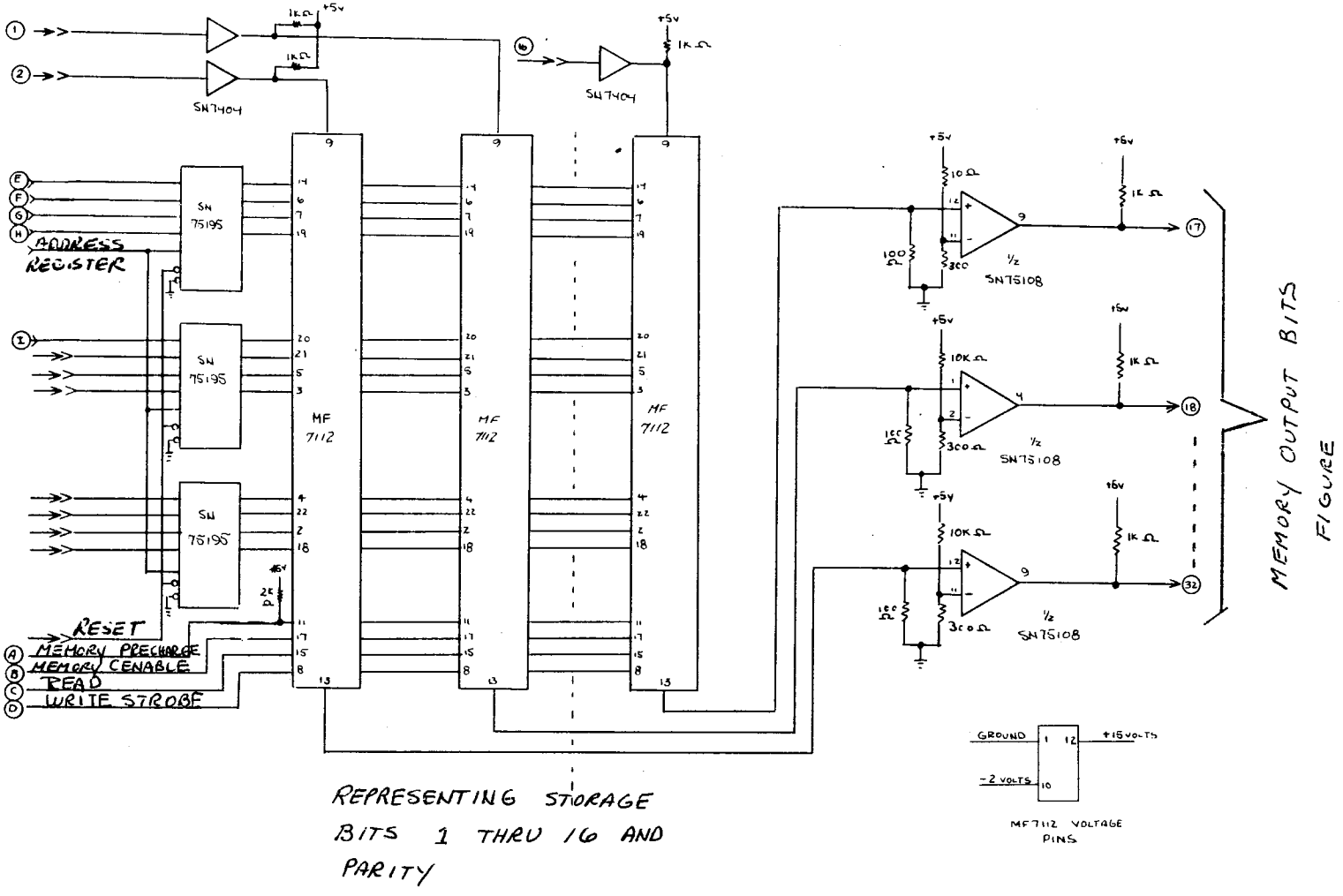


Figure 56. A Final Schematic of a Read/Write Memory Design Using Monostable Multivibrators for Interface Logic

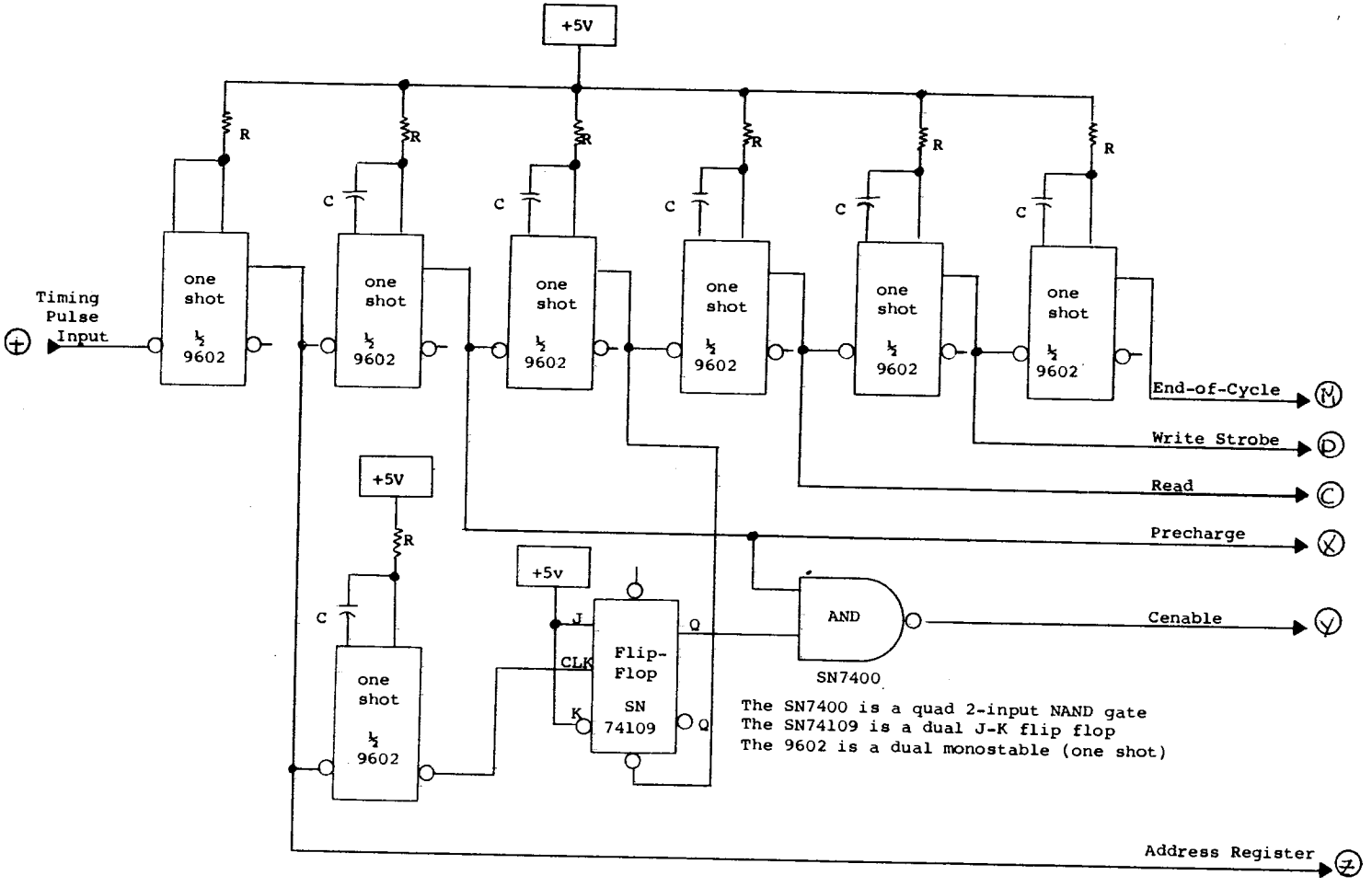


Figure 56. (Continued)

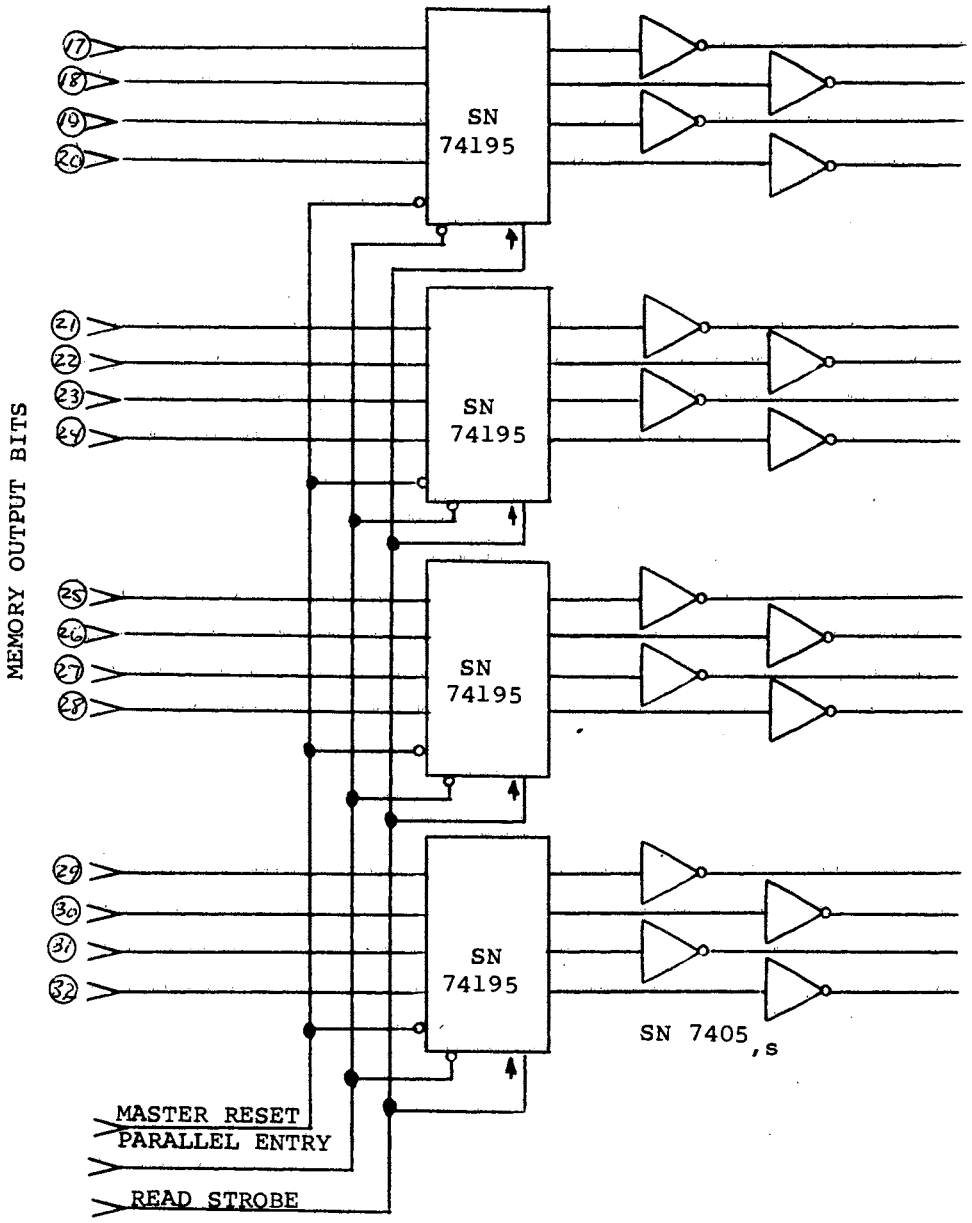


Figure 56. (Continued)

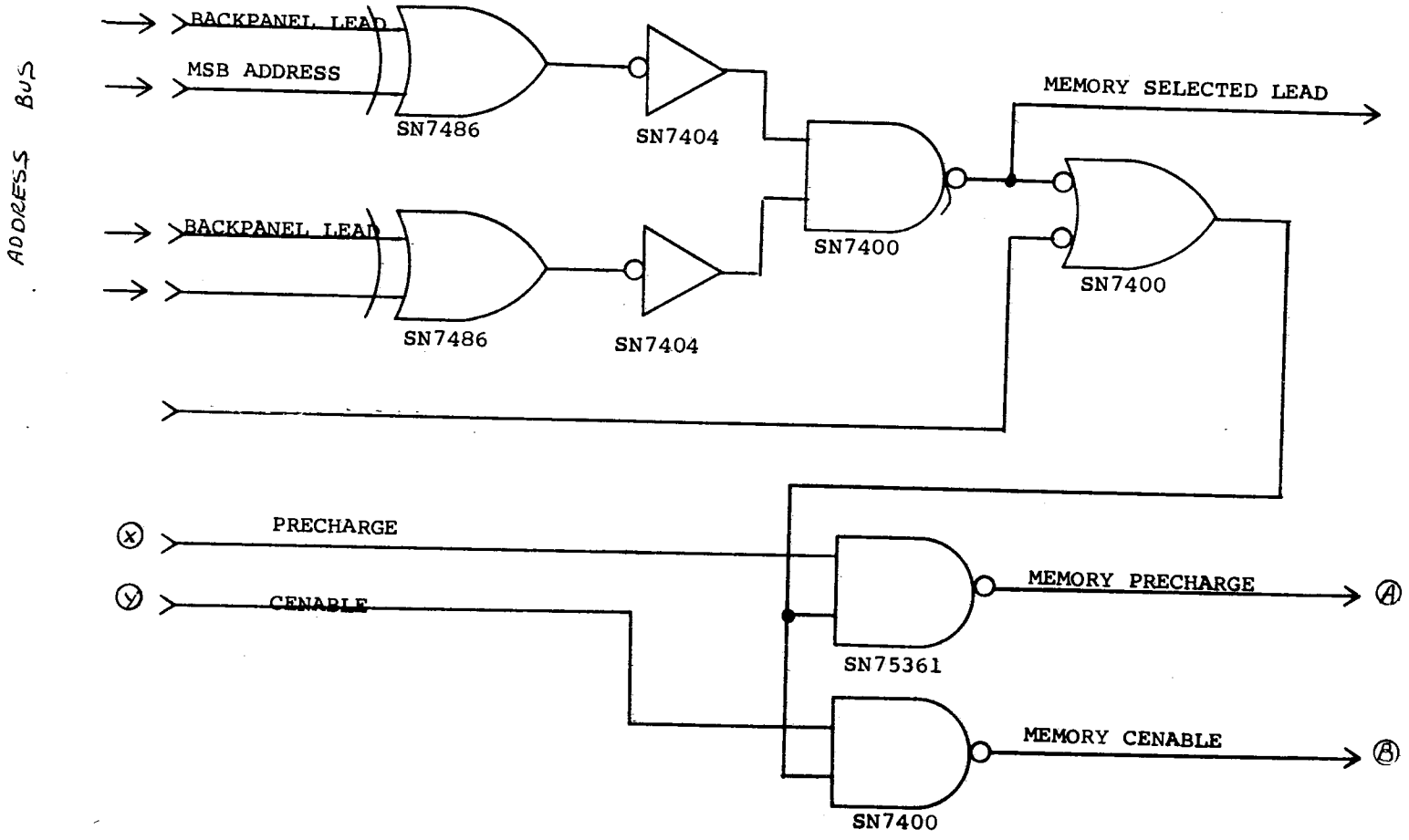


Figure 56. (Continued)

APPENDIX D

A FINAL SCHEMATIC OF A DESIGN USING A RING COUNTER INTERFACE LOGIC

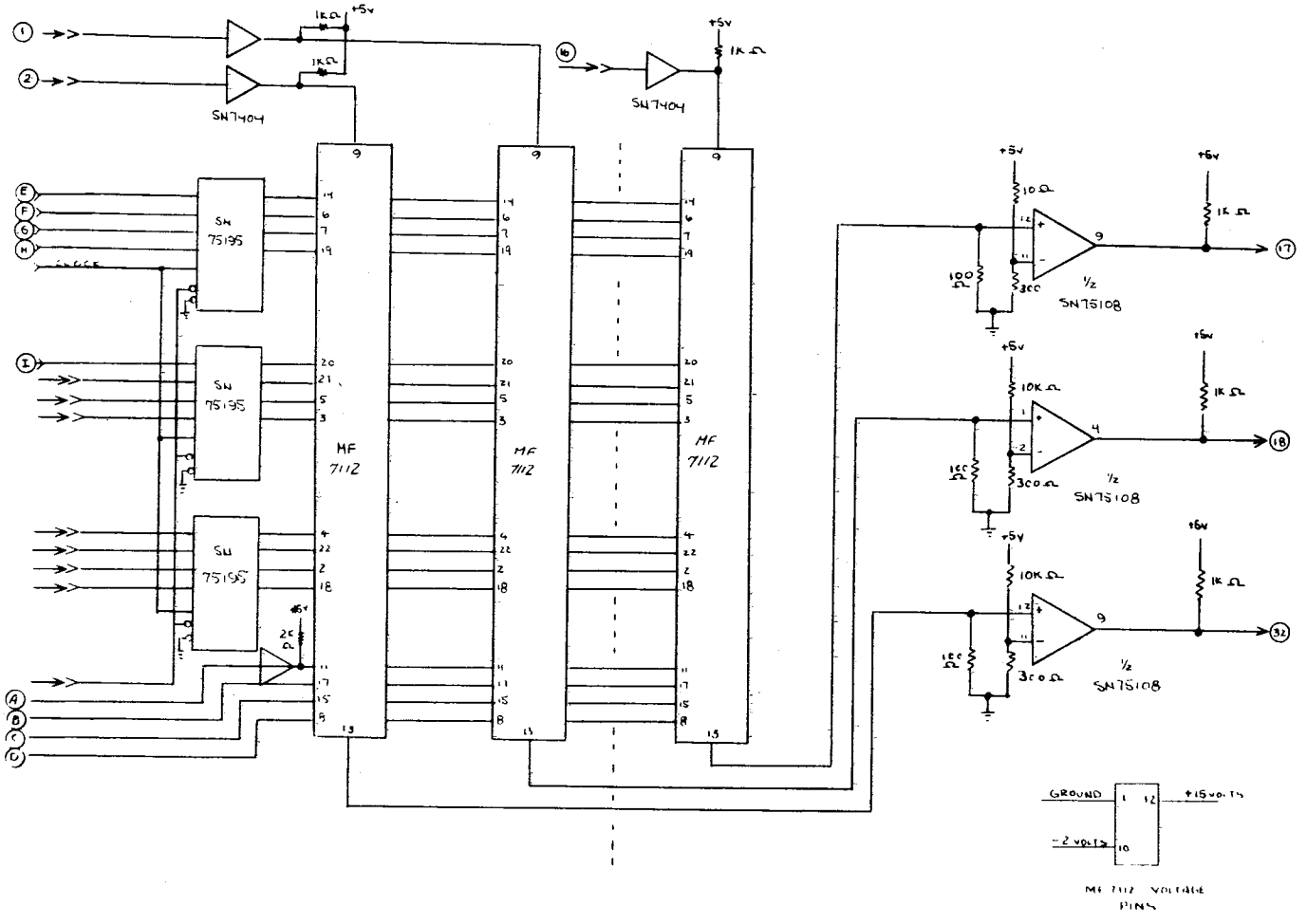


Figure 57. A Final Schematic of a Read/Write Memory Design Using a Ring Counter Interface Logic

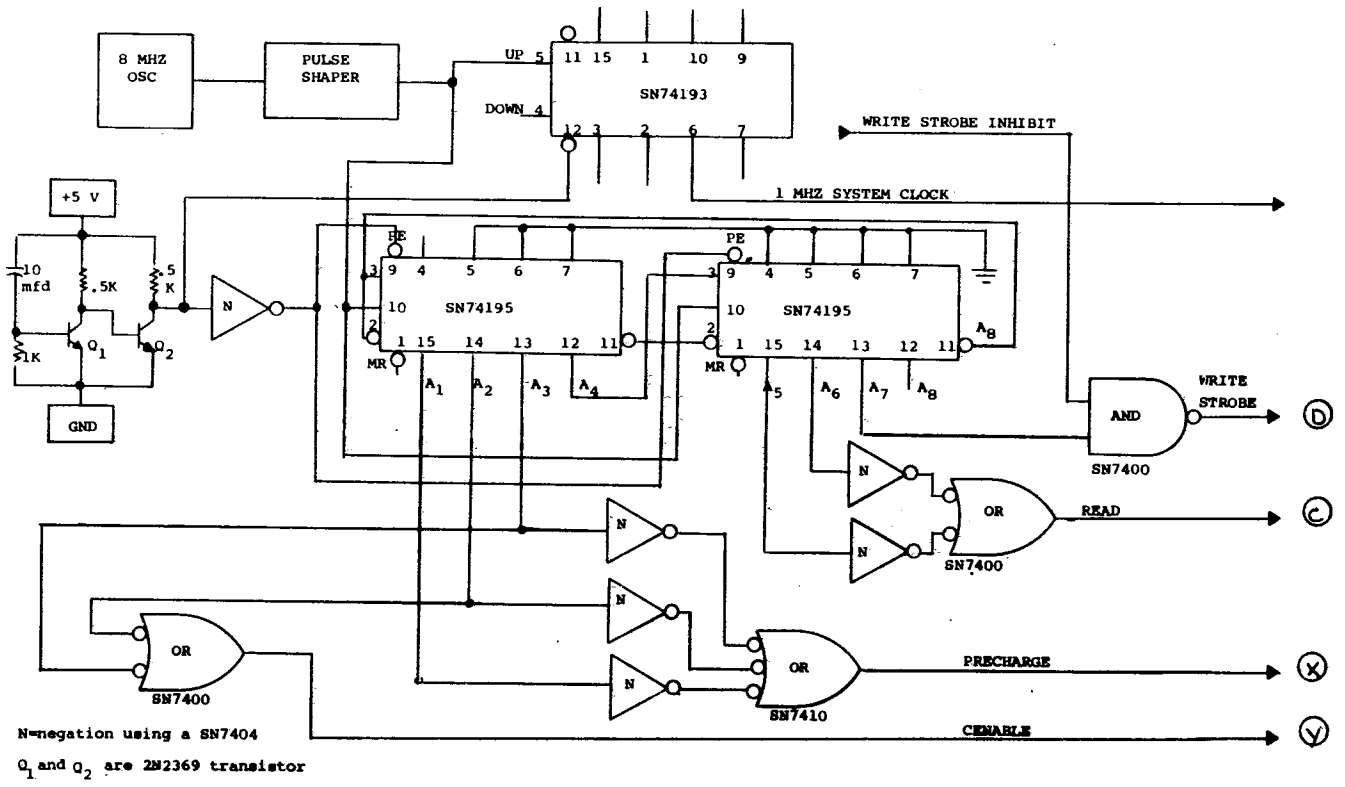


Figure 57. (Continued)

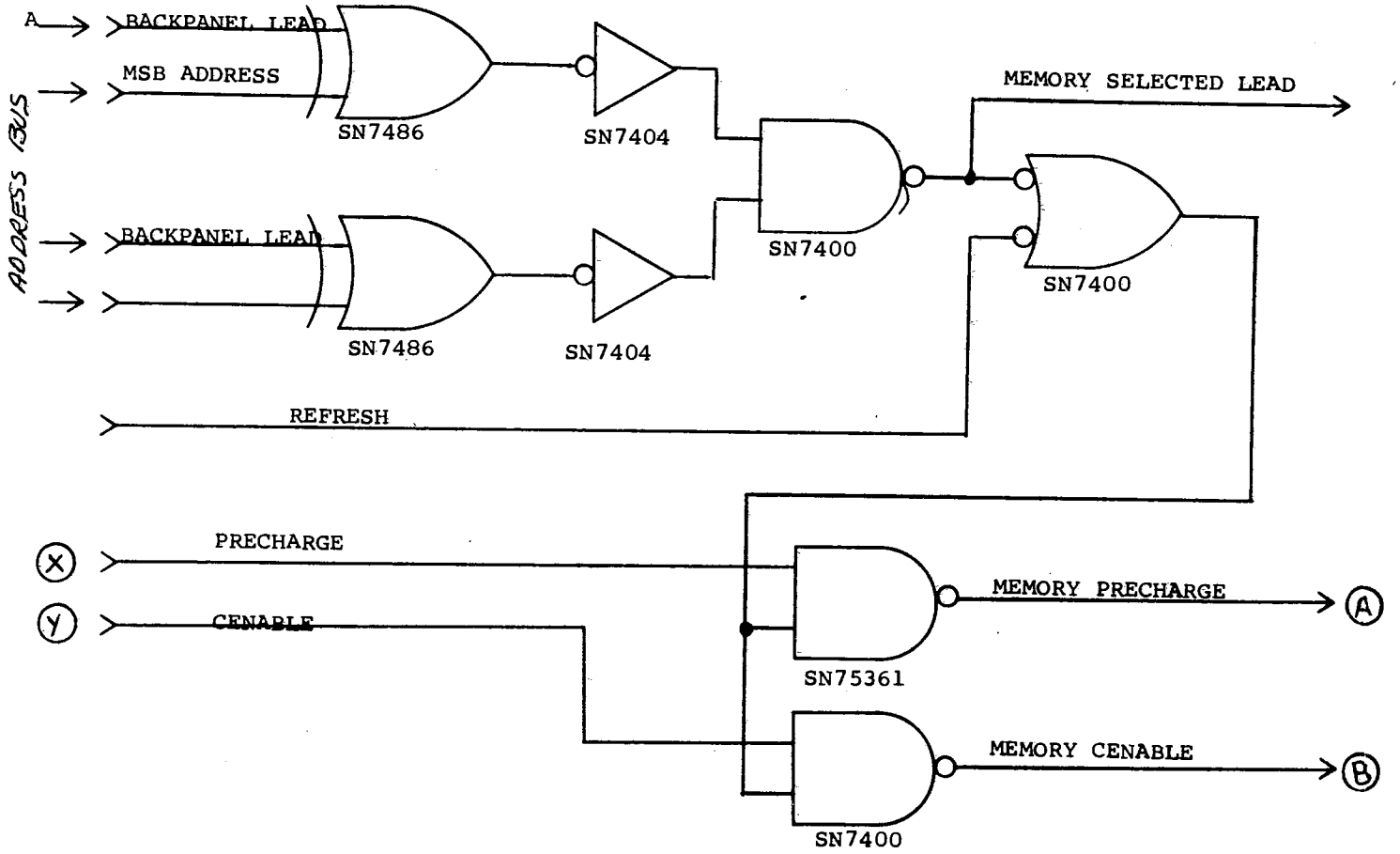


Figure 57. (Continued)

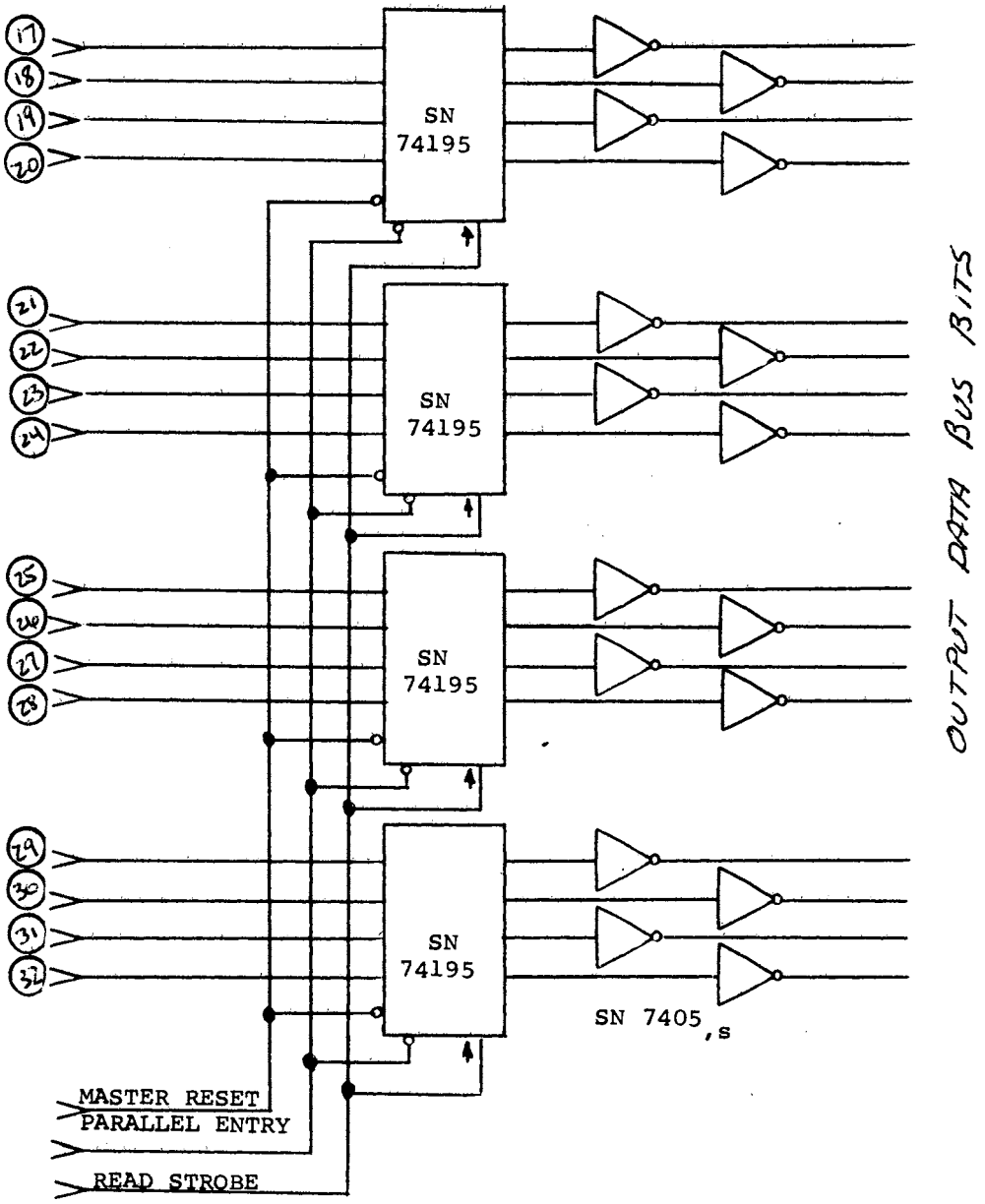


Figure 57. (Continued)

APPENDIX E

A FINAL SCHEMATIC OF THE READ ONLY RANDOM ACCESS MEMORY

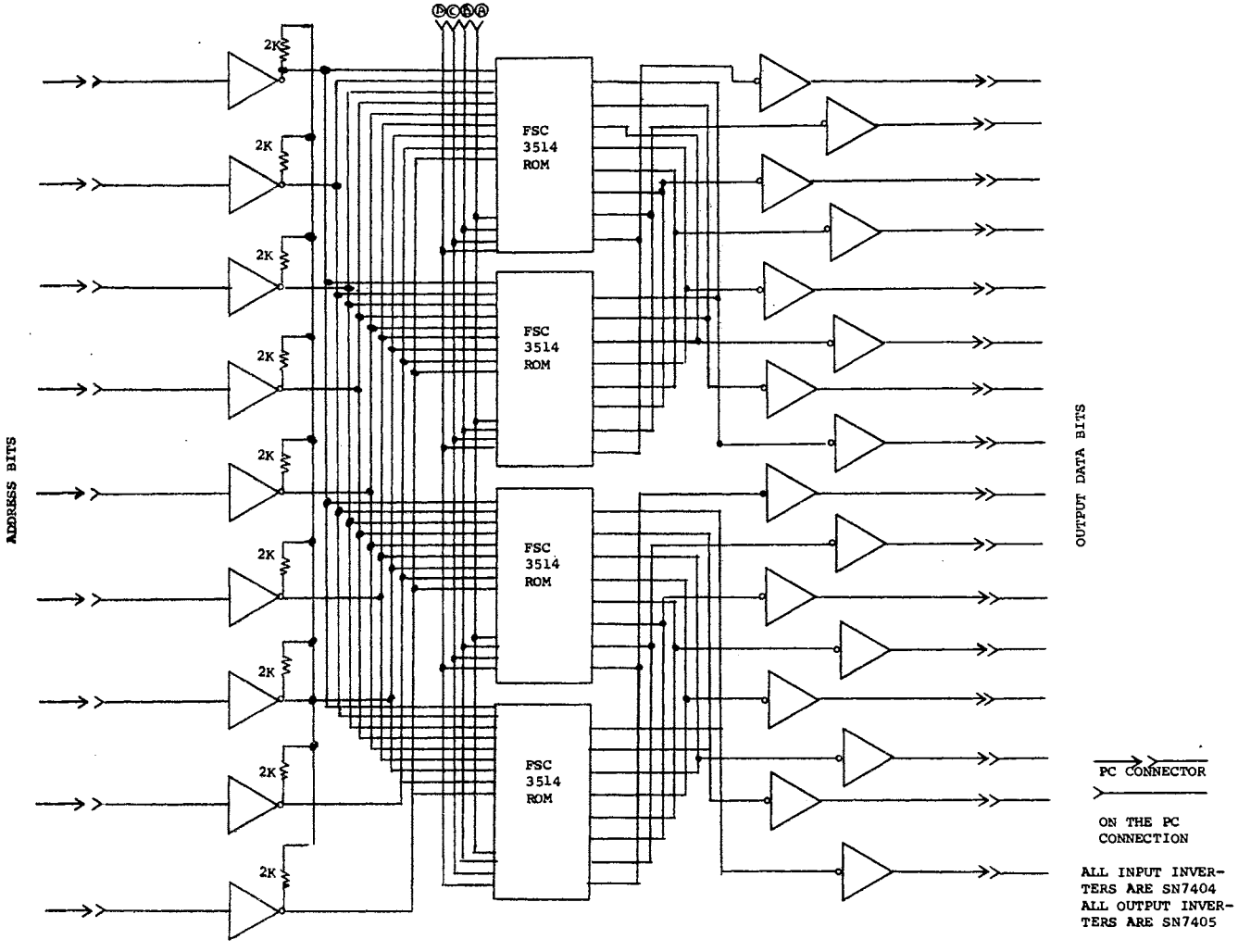


Figure 58. A Final Schematic of the Read Only Random Access Memory

APPENDIX F

THE MICROSYSTEMS INTERNATIONAL, LTD.

MF 7112, READ/WRITE IC DATA SHEET

Dec. 14, 1972

MICROSYSTEMS INTERNATIONAL, LTD.

MF7114 4K N-CHANNEL R.A.M. REV. 4

BLOCK DIAGRAM:

The memory is arranged in two arrays of 16 x 128 cells. Each array is further divided into four 16 x 32 arrays. This is done to minimize capacitive loading on critical nodes. Columns of 128 memory cells and 1 status cell are selected by the X decoder with addresses A_0 through A_4. Rows of 32 cells are selected by the Y decoder with addresses A_5 through A_{11}. Duplicate X decoders are used.

This enables greater drive to be applied to the read and write lines which appear on the layout as polysilicon stripes. To keep the number of read and write drive circuits to a minimum, four block data lines (512 cells per block) and their associated status cell line are multiplexed into each read/write circuit by a simple extra decode taken from A_{10} and A_{11}. This decode selects the correct one out of four Block data lines to be compared with the status cell line, and presents it to the exclusive NOR circuits which determine the state of data to be output from or written into the memory.

The array is split into two halves of 2K words x 1 bit each, the I/P and O/P being connected together to achieve the 4K by 1 organization. This chip organization is used to advantage in reducing the amount of time required to refresh the memory. By selecting one column (128 memory cells plus 1 status) from both halves of the memory at the same time, only the first four address pins need be cycled as each cycle refreshes 258 cells (256 memory + 2 reference cells). This is achieved by disabling the Y decoder and running both halves of the X decoder together, which is made possible by forcing A_5 and $\overline{A}_5$ to a "1" and A_4 and $\overline{A}_4$ to a "0" internally during a refresh operation.

DEVICE OPERATION:

Referring to the block diagram (Fig. 1) and the timing diagram (Fig. 2) it can be seen that the device cycle timing is established via the three clocks $\emptyset_1$ $\emptyset_2$ $\emptyset_3$. The cycle is split into three separate portions by these clocks and this enables three types of cycle operation to be performed: (1) Read Only (2) Read/Write (3) Read Modify Write. No read/write control line is used as $\emptyset_2$ and $\emptyset_3$ perform this function and the chip enable pulse is used only to select the required memory chips in use at any particular time.

REFRESH CYCLES:

Because of the dynamic storage nature of the 4K matrix the whole array must be periodically refreshed. The refresh cycle is a normal Read/Write cycle and to simplify the control of this cycle, and also speed up the total time taken to refresh the complete matrix, a refresh control is used. This control normally a "1" is taken to "0" for the complete $\emptyset_1$ $\emptyset_2$ $\emptyset_3$ cycle and is arranged to have preference over chip enable so that a refresh can be accomplished regardless of the state of chip enable. Simple gating on the X and Y decoders, also controlled from the refresh signal, enables a complete refresh to be performed in 16 memory cycles (i.e. the cycling of A_0 through A_3).

Data is automatically re-written every $\emptyset_1$, $\emptyset_2$, $\emptyset_3$ cycle, so no refresh amplifiers are needed as refresh is accomplished by the inverting cell itself. The refresh time is kept to a minimum, and in terms of lost memory cycles is only .43% of available read/write cycles and .37% of available read only cycles. Alternatively refresh can be accomplished during normal memory use. This is achieved by exercising each column of 128 memory cells plus 1 status cell at least once every 2mS. If refresh is done this way 32 cycles are needed for a complete refresh, but as these are normal Read/Write or Read-Modify-Write cycles, no memory time is lost.

INITIALIZATION OF STATUS CELLS

During operation of the memory, the function of the status cells is to maintain a constant phase relationship with respect to the memory cells in the associated column. However, during the first few cycles it could be possible for the status cell to lose this relationship because of initial set up conditions in the memory. This could be overcome by cycling the memory a number of times before using it each time a power-on sequence occurred. This is time consuming, and a much simpler and quicker way is to perform a refresh cycle with $\emptyset_2$ inhibited. This way the precharge condition is written into each memory and status cell position a column at a time. This operation requires the usual 16 cycles required for a normal refresh. It is also possible to use this as a clear memory operation.

It would also be possible to achieve the same result by overlapping the clocks $\emptyset_3$ and $\emptyset 1$ for a shor time. This would leave the write line at a "1" while charging all the data lines, thereby writing the charged data line onto each storage node. This is a one shot operation as $\emptyset_1$ also has the effect of setting all the X and Y decoders to a "1". Therefore instead of initializing cells a column at a time, all cells are initialized at one time.

REV. 4 DEC 14, 1972

Maximum Ratings

Maximum voltage applied to any or all input pins with respect to pin 10 is:	+25V	-0.3V
V_{DD} (with respect to ground)	+15V	-0.3V
V_{BB} (with respect to ground)	-5V	+0.3V
Operating temperature range	0°C to	70°C
Storage Temperature range	-55°C to	+125°C
Typical power dissipation	500 mW	

Electrical Characteristics

T_A = 25°C, V_{DD} = +0.0 ± 1.0V, V_{BB} = -2.0 ± 0.2V

		MIN	TYP.	MAX.	UNITS
Clock Inputs	Ø1	+16			V
	High State Ø2, Ø3	+12			V
	Low State	-0.3		.4	V
All Other Inputs	High State	+2.0		+10	V
	Low State	-0.3		+0.8	V
Input current Ø1 Clock (V_{in}=+16V)			40		mA
Ø2, Ø3 clocks (V_{in} =+15V)				2	mA
All Other Inputs (V_{in} +15V)				1.0	uA
Output Sink Current	High State 05V			100	uA
	Low State 03V	3.0			mA
Input Capacitance	Ø1 Clock		80 (0+5.0V)	120 (0. 0.0V)	pf
with V_{BB} = -2V	Ø2, Ø3 Clocks		27 (0 +5.0V)	35 (0 0.0V)	pf
	All other Inputs		5 (0 0.0V)	7 (0 0.0V)	pf
Output Capacitance With V_{BB} + -2V			6.0 (0 +5.0V)	8.0 (0 +5.0V)	pf

Dec. 14, 1974

Electrical Characteristics (continued)

		MIN.	TYP.	MAX.	UNITS
Power Supply Current (average)	V_{DD}		15		mA
	V_{BB}			1	mA
(NOTE: T_1-T_7 are measured at the 10% level) Precharge Width	T_1	300			nS
Precharge-Read	T_2	100			nS
Read Width	T_3	180			nS
Read Write	T_4	0			nS
Write Width	T_5	100			nS
Write Precharge	T_6	0			nS
Address End of Precharge	T_7	200			nS
Address Q/P strobe (Access Time)	TA_{CC}			490	nS
Read Cycle				580	nS
Read/Write Cycle				680	nS

BLOCK DIAGRAM

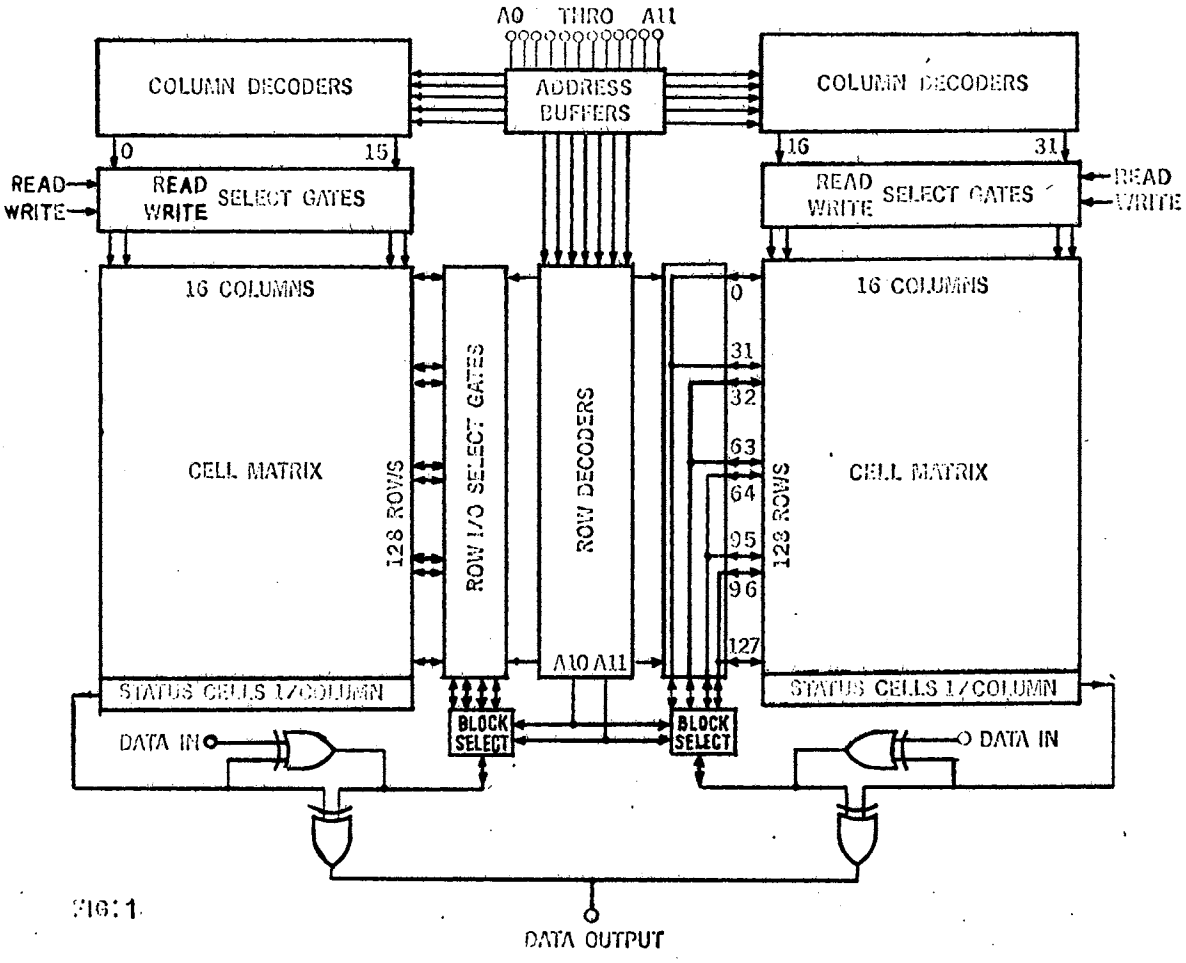
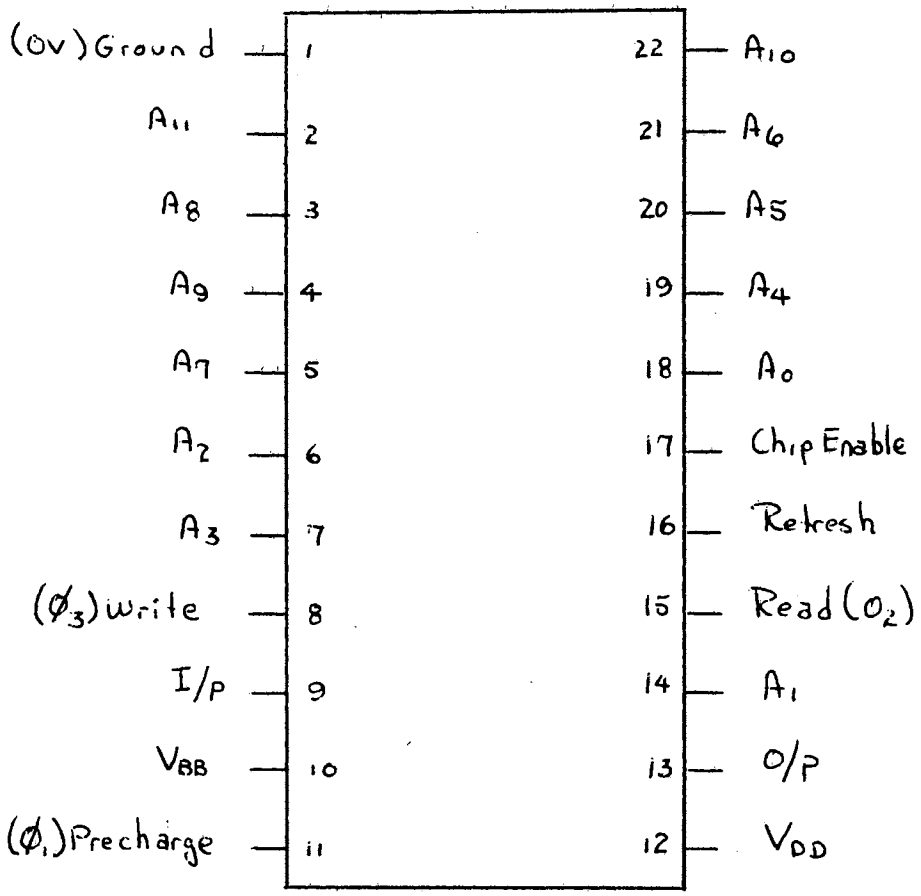


FIG: 1


(0v) Ground
1
A_{11}
2
A_8
3
A_9
4
A_7
5
A_2
6
A_3
7
(ϕ_3) write
8
I/P
9
V_{BB}
10
(ϕ_1) Precharge
11
22
A_{10}
21
A_6
20
A_5
19
A_4
18
A_0
17
Chip Enable
16
Refresh
15
Read (O_2)
14
A_1
13
O/P
12
V_{DD}

APPENDIX G

THE FAIRCHILD SEMICONDUCTOR, INC. 3514

READ ONLY MEMORY DATA SHEET

3514
4096-BIT READ ONLY MEMORY
FAIRCHILD SILICON GATE MOS INTEGRATED CIRCUIT

GENERAL DESCRIPTION – The 3514 is a 4096-Bit Read Only Memory. It is organized in a 512-word by 8-bit format. There are four programmable chip selects to provide for OR wiring up to 16 chips. The 3514 is an MOS monolithic integrated circuit utilizing P-channel enhancement mode silicon gate technology.

- **INPUT AMPLIFIER ELIMINATES ALL PULL UP RESISTORS**
- **INTERFACE DIRECTLY WITH TTL – NO EXTERNAL COMPONENTS**
- **500 ns TYPICAL ACCESS TIME**
- **4-BIT PROGRAMMABLE CHIP SELECT CODE**
- **WIRED-OR CAPABILITY ON OUTPUTS**
- **STATIC – NO CLOCKS REQUIRED**

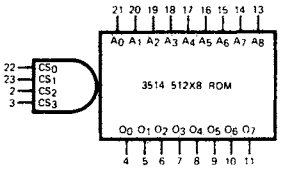
ABSOLUTE MAXIMUM RATINGS

Storage Temperature	−65°C to +150°C
Operating Temperature (T_A)	0°C to +70°C
Voltage on Any Input Pin (V_{SS} = GND)	−20 V to +0.3 V
Voltage on V_{DD} (V_{SS} = GND)	−7 V to +0.3 V
Voltage on Output Pin (V_{SS} = GND, Output Current @ ±10 mA)	−7 V to +0.3 V

APPLICATIONS

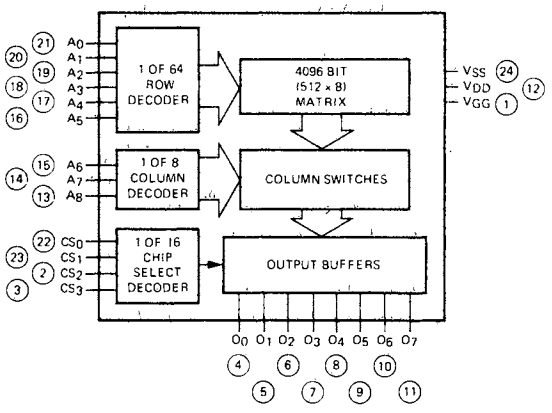
Code Conversion
Microprogramming
Table Lookup
Control Logic

LOGIC SYMBOL



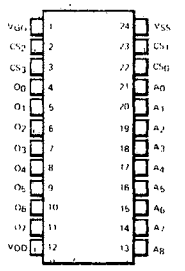
Pin 24 = V_{SS}
Pin 12 = V_{DD}
Pin 1 = V_{GG}

LOGIC BLOCK DIAGRAM



○ = Pin Number

CONNECTION DIAGRAM
(TOP VIEW)



FAIRCHILD MOS INTEGRATED CIRCUIT • 3514

FUNCTIONAL DESCRIPTION – A 9-bit binary address applied to the address inputs (A_0–A_8) will cause a corresponding 8-bit word to appear on the outputs (O_0–O_7). A 4-bit programmable chip select (CS_0–CS_3) allows selection of one-of-sixteen memories without external gating. When a chip is not selected, its outputs are turned off, i.e., a high impedance to both V_{SS} and V_{DD}. This feature allows expansion of up to 16 memories without external components, either for chip select decoding or for output gating. Each output of the device will drive 1.5 unit TTL loads, either one standard TTL device and two low power TTL devices or six low power devices.

Each input to the 3514 drives a special input amplifier stage which eliminates all pull up resistors (either on or off chip).

Two types of information are to be supplied when ordering the 3514. First, the bit pattern to be stored in the 512-word locations of memory, and second, the 4-bit chip enable code which will activate the chip.

DC CHARACTERISTICS: V_{GG} = -12 V ±5%, V_{DD} = 0 V, V_{SS} = ±5%, T_A = 0°C to +70°C

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNIT	CONDITIONS
V_{IH}	Input Voltage HIGH	V_{SS}–2.75 V		V_{SS}	V	
V_{IL}	Input Voltage LOW	V_{GG}	0	0.55	V	
V_{OH}	Output Voltage HIGH	2.4		V_{SS}	V	I_{OH} = 0.5 mA
V_{OL}	Output Voltage LOW	0		0.4	V	I_{OL} < 2.4 mA
I_{LI}	Input Leakage Current			1.0	μA	V_{IN} = V_{SS} –6 V, Note 1
I_{LO}	Output Leakage Current			1.0	μA	V_{OUT} = V_{SS} –6 V, Note 2
I_{DD}	V_{DD} Current		19	25	mA	
I_{GG}	V_{GG} Current		19	25	mA	
I_{SS}	V_{SS} Current		38	50	mA	
P_D	Power Dissipation		450	580	mW	

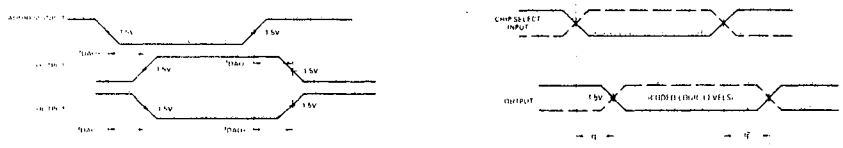
NOTES: 1. All pins at 0 V except those under test.
2. Output floating (chip not selected)

AC CHARACTERISTICS: V_{GG} = -12 V ±5%, V_{DD} = 0 V, V_{SS} = ±5 V ±5%, T_A = 0°C to +70°C

SYMBOL	CHARACTERISTIC	MIN.	TYP.	MAX.	UNITS	CONDITIONS
t_{DAO+}	Access Time Address to Output HIGH					Notes 1 and 2
	35141		500	700	ns	
	35142		.600	1.0	μs	
t_{DAO-}	Access Time Address to Output LOW					Notes 1 and 2
	35141		500	700	ns	
	35142		.600	1.0	μs	
t_E	Access Time Chip Select Enable to Output					Note 2
	35141		450	500	ns	
	35142		750	900		
$t_{\bar{E}}$	Access Time Chip Select Disable to Output					Note 2
	35141		550	600	ns	
	35142		750	900		
C_I	Input Capacitance		3.0	8.0	pF	
C_O	Output Capacitance		7.0	12	pF	Note 3

NOTES: 1. 1.5 TTL load
2. See timing diagram and characteristic curves.
3. Output floating (chip not selected)

TIMING DIAGRAM



TYPICAL ELECTRICAL CHARACTERISTICS

Fig. 1
ACCESS TIME VERSUS SUPPLY VOLTAGE

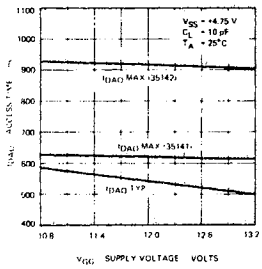


Fig. 2
ACCESS TIME VERSUS SUBSTRATE VOLTAGE

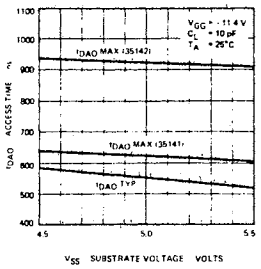


Fig. 3
ACCESS TIME VERSUS AMBIENT TEMPERATURE

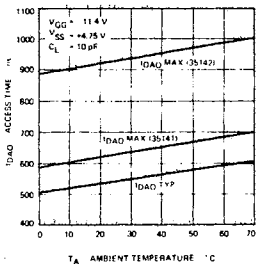


Fig. 4
ACCESS TIME VERSUS LOAD CAPACITANCE

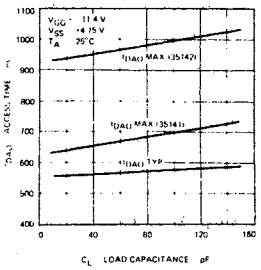


Fig. 5
V_{GG} CURRENT VERSUS AMBIENT TEMPERATURE

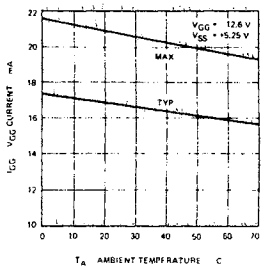


Fig. 6
V_{DD} CURRENT VERSUS AMBIENT TEMPERATURE

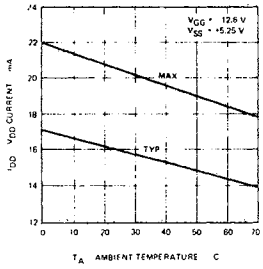


Fig. 7
V_{GG} CURRENT VERSUS SUPPLY VOLTAGE

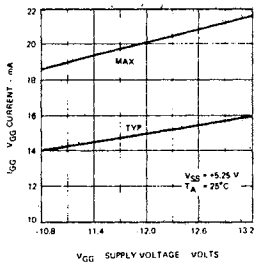
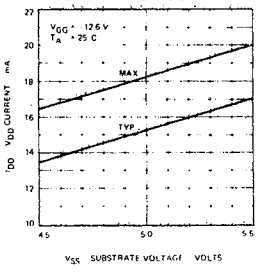


Fig. 8
V_{DD} CURRENT VERSUS SUBSTRATE VOLTAGE



FAIRCHILD MOS INTEGRATED CIRCUIT • 3514

CUSTOM PATTERN ORDERING INFORMATION – The 3514 is programmed on IBM cards or IBM coding form in the format as shown below. (Specify access time desired on Purchase Order).

Logic "1" = a more positive voltage (normally +5 V)
Logic "0" = a more negative voltage (normally o V)

FIRST CARD

Column Number	Description
10 thru 29	Customer Name
50 thru 62	35141 or 35142
65 thru 72	Date

SECOND CARD

Column Number	Description
29	CS_3 input required to select chip
31	CS_2 input required to select chip
33	CS_1 input required to select chip
35	CS_0 input required to select chip

REMAINING 512 CARDS

Column Number	Description
10, 12, 14, 16, 18, 20, 22, 24, 26	Address input pattern. The most significant bit (A_8) is in column 10
40, 42, 44, 46, 48, 50, 52, 54	Output pattern. The most significant bit (O_7) is in column 40
73 through 80	Coding these columns is not essential and may be used for card identification purposes.

GLOSSARY OF TERMS

1. V_{SS}: The most positive voltage applied to the device.
2. V_{GG}: The most negative voltage applied to the device.
3. V_{DD}: The next most negative voltage applied to the device.
4. Address access time:
   - t_{DAO+} The time delay from an address input logic HIGH or logic LOW state to an output logic HIGH state.
   - t_{DAO-} The time delay from an address input logic HIGH or logic LOW state to an input logic LOW state.
5. t_E Chip enable time: The time delay from Valid Code at chip select inputs to an output logic HIGH or logic LOW state.
6. $t_{\bar{E}}$ Chip disable time: The time delay from removal of a valid chip select input code to a high state on outputs when driving a standard TTL load.
7. T_A: Still air ambient temperature.

VITA

Robert P. Billeg, Jr.

Candidate for the Degree of

Master of Science

Thesis: ADAPTING SEMICONDUCTOR MEMORY TO SMALL MACHINE APPLICATIONS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Bloomsburg, Pennsylvania, December 20, 1938, the son of Mr. and Mrs. R. P. Billeg.

Education: Graduated from Central High School, Oklahoma City, Oklahoma, in May, 1956; received the Bachelor of Science degree in Electrical Engineering from the University of Oklahoma in 1963; enrolled in the master's program at the Oklahoma State University Extention Center in 1968; completed requirements for the Master of Science degree at Oklahoma State University in May, 1974.

Professional Experience: Project and Evaluation Engineer, United States Air Force, 1963-1965; member of the General Electric, Incorporated, Oklahoma City, design engineering staff, 1965-1970; member of the Honeywell Informations Systems, Incorporated, design engineering staff, 1970 to present.